

**PROTOTIPE SENSOR PARKIR MOBIL
BERBASIS MIKROKONTROLER AT89S51**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Mencapai Gelar Ahli Madya
Program Diploma III Ilmu Komputer



Diajukan Oleh :

MUHAMAD YUSUF
NIM. M3306023

PROGRAM DIPLOMA III ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET
SURAKARTA
2009

ABSTRACT

Muhamad Yusuf, 2009. **Car parking sensor prototype based on AT89S51 microcontroller**. D3 Programs Computer Science Faculty of Mathematics and Natural Sciences University of Eleven March.

Car drivers often experience difficulties to park their cars in the narrow location, because the parking area is narrower. Not a few drivers hit the power pole or deface the walls when parking the car.

The goal of this final task is to design a car parking sensor prototype based on AT89S51 microcontroller. This tool system uses a AT89S51 microcontroller as a main control. Used as input SRF04 ultrasonic sensor module which, it consists TX (transmitter) and RX (receiver). Output use an LCD (Liquid Crystal Display) to display the distance and uses speakers for sound indicators.

Results obtained from that car parking sensor prototype based on AT89S51 microcontroller is a tool can be used to measure the distance of parking. AT89S51 microcontroller functions is the main data processing equipment on the resulting distance from ultrasonic sensor. Use module SRF04 ultrasonic sensor, the car parking sensor prototype based on AT89S51 microcontroller can produce a more accurate data. Parking distance is presented via LCD display with a measuring meter. Speaker can be used as sound indicators on the car parking sensor.

Keywords: parking sensor, SRF04 ultrasonic sensor, AT89S51 microcontroller

INTISARI

Muhamad Yusuf, 2009. **PROTOTIPE SENSOR PARKIR MOBIL BERBASIS MIKROKONTROLER AT89S51**. Program Studi D3 Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta.

Pengemudi mobil seringkali mengalami kesulitan untuk memparkir mobilnya di lokasi sempit, disebabkan lahan parkir yang semakin berkurang. Tidak sedikit pengemudi yang menabrak tiang listrik atau menggores tembok ketika memundurkan mobilnya.

Tujuan dari tugas akhir ini adalah untuk merancang Prototipe Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51. Sistem alat ini menggunakan mikrokontroler AT89S51 sebagai pengendali utamanya. Sebagai *input* digunakan modul sensor ultrasonik SRF04 yang terdiri TX (*transmitter*) dan RX (*receiver*). Sebagai *output* digunakan sebuah LCD (*Liquid Crystal Display*) untuk menampilkan jarak dan *speaker* untuk indikator bunyi.

Hasil yang didapat dari pembuatan prototipe sensor parkir mobil berbasis mikrokontroler AT89S51 adalah alat tersebut dapat digunakan untuk mengukur jarak parkir. Mikrokontroler AT89S51 berfungsi sebagai pengendali utama pada pemrosesan data jarak parkir yang dihasilkan dari sensor ultrasonik. Penggunaan modul sensor ultrasonik SRF04 dapat menghasilkan data yang lebih akurat. Jarak parkir ditampilkan melalui LCD dengan satuan ukur meter. Speaker dapat digunakan sebagai indikator suara pada sensor parkir mobil.

Kata kunci : *Sensor parkir, sensor ultrasonik SRF04, mikrokontroler AT89S51*

MOTTO

- ❖ **Hari Ini Harus Lebih Baik Dari Hari Kemarin**
- ❖ **Ingatlah, Anda menjadi apapun yang teratur Anda lakukan
(Mario Teguh)**

PERSEMBAHAN

Kupersembahkan karya ini untuk :

☞ ***Bapak dan Ibu ku yang tercinta.***

Terima kasih atas do'a dan dorongan yang tiada henti.

☞ ***Kakak ku yang tercinta. Terima kasih atas segala bantuannya.***

☞ ***Yayan, Aji, Wahyu, Sigit, N'g.***
Ma kasih bantuannya.

☞ ***Teman – teman D3 TEKNIK***
KOMPUTER angkatan 2006 yang
selalu memberi motivasi dan
semangat.

☞ ***Teman – teman L.E yang selalu santai***

KATA PENGANTAR

بسم الله الرحمن الرحيم

Puji syukur penyusun panjatkan ke hadirat Allah SWT, atas limpahan rahmat dan hidayah-Nya, sehingga penyusun dapat melaksanakan tugas akhir dan menyusun laporan tugas akhir yang berjudul **“PROTOTYPE SENSOR PARKIR MOBIL BERBASIS MIKROKONTROLER AT89S51”** ini dengan sebaik-baiknya.

Laporan tugas akhir ini disusun sebagai pelengkap salah satu syarat mencapai gelar Ahli Madya Program Diploma III Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta.

Penulis mengucapkan terima kasih dan memberikan penghargaan yang sebesar-besarnya kepada :

1. Bapak Prof. Drs. Sutarno, M.Sc, Ph.D, selaku Dekan FMIPA UNS.
2. Bapak Drs YS. Palgunadi, M.Sc, selaku ketua jurusan DIII Ilmu Komputer FMIPA UNS.
3. Bapak Drs. Syamsurizal selaku dosen pembimbing yang telah membantu dan membimbing sehingga selesai tugas akhir ini.
4. Bapak dan Ibu serta kakakku, penulis ucapkan banyak terima kasih atas bantuan dan do'anya.
5. Teman-teman seperjuangan **”D3 Teknik Komputer Universitas Sebelas Maret angkatan 2006”** yang telah memberi semangat dan bantuan pada penulis.

Semua pihak yang telah membantu baik materiil maupun spiritual yang tidak dapat penyusun sebutkan satu persatu.

Penulis menyerahkan semua proses penulisan tugas akhir ini hanya kepada Allah Ta'ala dan Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Surakarta, 1 Mei 2009

Penyusun

DAFTAR ISI

JUDUL	I
HALAMAN PERSETUJUAN	II
HALAMAN PENGESAHAN	III
ABSTRAK	IV
INTISARI	V
MOTTO	VI
PERSEMBAHAN	VII
KATA PENGANTAR	VIII
DAFTAR ISI	IX
DAFTAR TABEL	XII
DAFTAR GAMBAR	XIII
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	1
1.3 Batasan Masalah	1
1.4 Tujuan dan Manfaat	2
1.4.1 Tujuan	2
1.4.2 Manfaat	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Laporan	3
BAB II LANDASAN TEORI	4
2.1 Gambaran Umum Mikrokontroler	4
A. <i>Central Processing Unit</i> (CPU)	5
B. <i>Read Only Memory</i> (ROM)	5
C. <i>Random Acces Memory</i> (RAM)	5
D. <i>Input / Output</i> (I/O)	5
E. Komponen lainnya	5
2.2 Mikrokontroler AT89S51	6

2.2.1	Arsitektur Mikrokontroler AT89S51.....	6
2.2.2	Memori Internal AT89S51	11
1.	<i>Read Only Memory</i> (ROM)	11
2.	<i>Random Access Memory</i> (RAM)	12
3.	<i>Special Function Register</i> (SFR)	12
2.2.3	Osilator dan <i>Clock</i>	12
2.2.4	Bahasa <i>Assembly</i> Mikrokontroler AT89S51	14
2.2.5	Instruksi Mikrokontroler AT89S51	15
2.3	Sensor	17
2.3.1	Sensor Ultrasonik	18
A.	Pengertian Gelombang Ultrasonik	18
B.	Pengertian Sensor Ultrasonik	19
C.	Sensor Jarak Ultrasonik Devantech SRF04	20
2.4	<i>Liquid Crystal Display</i> (LCD)	22
2.5	<i>Loudspeaker</i>	24
BAB III DESAIN DAN PERANCANGAN.....		26
3.1	Analisis Kebutuhan	26
3.1.1	<i>Hardware</i>	26
1.	Minimum Sistem AT89S51	26
2.	Sensor Ultrasonik	26
3.	<i>Display</i> LCD	26
4.	<i>Speaker</i>	26
3.1.2	<i>Software</i>	27
1.	Proteus 7 Profesional	27
2.	Program <i>compiler</i> ASM51 dan AEC ISP	27
3.2	Diagram Alir Sensor Parkir Mobil.....	28
3.2.1	Diagram Alir Kalkulasi Data	29
3.3	Rancangan Skematik.....	30
1.	Diagram Blok Prototipe Sensor Parkir Mobil	30
2.	Desain Rangkaian Prototipe Sensor Parkir Mobil	30
3.	Catu daya	33

4. Rangkaian Modul Sensor Ultrasonik SRF04	33
5. Rangkaian <i>Display</i>	33
6. Rangkaian <i>Speaker</i>	34
3.3.1 Mencetak PCB	34
3.4 Tahap Penyelesaian.....	35
1. Merangkai komponen elektronik	35
2. Memasang PCB ke dalam box	35
3. Pemrograman mikrokontroler AT89S51	35
4. <i>Finishing</i>	38
5. Ujicoba	38
BAB IV IMPLEMENTASI DAN ANALISA	39
4.1 Uji Coba Alat	39
4.1.1 Pengujian Rangkaian Catu Daya	39
4.1.2 Pengujian Rangkaian Mikrokontroler	40
4.1.3 Pengujian Rangkaian Sensor Ultrasonik SRF04	43
4.1.4 Pengujian Rangkaian LCD	45
4.1.5 Pengujian <i>Speaker</i>	46
4.1.6 Pengujian Rangkaian Keseluruhan	47
BAB V PENUTUP	53
5.1 Kesimpulan	53
5.2 Saran	53
DAFTAR PUSTAKA	54
LAMPIRAN	55
1. Dokumentasi Alat	
2. Listing Program	
3. Datasheet Sensor Ultrasonik SRF04	
4. Datasheet LCD 2x16 SLCDV 1.2	

DAFTAR TABEL

Tabel 2.1 Fungsi Khusus Port 3	9
Tabel 2.2 Fungsi Pin LCD	23
Tabel 4.1 Hasil Pengukuran Jarak Dengan Rangkaian SRF04.....	43
Tabel 4.2 Hasil Perbandingan Pengukuran Jarak	44
Tabel 4.3 Hasil Pengujian Alat	48

DAFTAR GAMBAR

Gambar 2.1 Susunan Mikrokontroler	4
Gambar 2.2 Susunan Pin AT89S51	7
Gambar 2.3 Rangkaian <i>Clock</i>	13
Gambar 2.4 Diagram Blok AT89S51	13
Gambar 2.5 Proses Pemantulan Gelombang Ultrasonik	18
Gambar 2.6 Bentuk Fisik Sensor Ultrasonik Model Biasa	19
Gambar 2.7 Bentuk Fisik SRF04	21
Gambar 2.8 Bentuk Fisik LCD 2x16 Karakter	22
Gambar 2.9 Bentuk Fisik <i>Loudspeaker</i>	25
Gambar 3.1 <i>Flowchart</i> Cara Kerja Sensor Parkir	28
Gambar 3.2 <i>Flowchart</i> Kalkulasi Data	29
Gambar 3.3 Diagram Blok Prototipe Sensor Parkir Mobil	30
Gambar 3.4 Desain Rangkaian Prototipe Sensor Parkir Mobil	32
Gambar 3.5 Rangkaian Regulator 5V/500mA DC	33
Gambar 3.6 Rangkaian Modul Sensor Ultrasonik SRF04	33
Gambar 3.7 Rangkaian <i>Display</i> LCD	33
Gambar 3.8 Rangkaian <i>Speaker</i>	34
Gambar 3.9 <i>Load</i> Program yusuf.asm	36
Gambar 3.10 File yusuf.asm	36
Gambar 3.11 File yusuf.hex	36
Gambar 3.12 Proses <i>Load File</i> Hex	37
Gambar 3.13 Proses Inisialisasi Memori Program	37
Gambar 3.14 <i>Download</i> Program ke IC AT89S51	38
Gambar 3.15 Proses <i>Download</i> Program ke IC Berhasil	38
Gambar 4.1 Pengujian Rangkaian Catu Daya.....	40
Gambar 4.2 Pengujian IC AT89S51	41
Gambar 4.3 Pengujian Rangkaian Sensor Ultrasonik SRF04	42
Gambar 4.4 Pengujian Rangkaian LCD	45
Gambar 4.5 Pengujian Speaker	46

Gambar 4.6 Prototipe Sensor Parkir Mobil.....	47
---	----

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pengemudi kendaraan roda empat seringkali mengalami kesulitan untuk memparkir mobilnya di lokasi sempit, disebabkan lahan parkir yang semakin berkurang. Tidak sedikit pengemudi yang menabrak tiang listrik atau menggores tembok ketika memundurkan mobilnya, hal ini disebabkan karena pengemudi tidak mengetahui kondisi di belakang kendaraan yang ditumpangnya karena keterbatasan pandangan.

Kondisi gelap juga menjadi salah satu penyebab terjadinya benturan di bumper belakang. Beberapa orang menyiasatinya dengan memasang *rear* ban tambahan pada bumper belakang untuk mengurangi kerusakan akibat benturan.

Hal inilah yang mendorong untuk merancang Prototipe Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51. Prototipe sensor parkir mobil ini akan memberikan informasi berupa indikator suara dari *speaker* (pengeras suara) dan dilengkapi dengan LCD untuk menampilkan jarak antara bumper belakang dengan benda yang ada dibelakang mobil.

1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, maka dapat diambil perumusan masalah yaitu bagaimana merancang atau membuat Prototipe Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51.

1.3 Batasan Masalah

Karena luasnya materi, maka dilakukan beberapa pembatasan masalah, antara lain :

- a. Mikrokontroler yang digunakan adalah AT89S51.
- b. Menggunakan bahasa pemrograman ASM (*assembler*).

- c. Prototipe yang dibuat menggunakan sensor ultrasonik SRF04 yang dapat mengukur jarak.
- d. Menggunakan LCD dan *speaker* sebagai pemberi informasi jarak parkir.

Tujuan dibuatnya batasan masalah adalah agar pokok-pokok permasalahan yang dibahas tidak melenceng dari topik yang telah diangkat.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan dari tugas akhir ini adalah untuk merancang Prototipe Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51 dengan menggunakan sensor ultrasonik sebagai pendeteksi jarak parkir mobil.

1.4.2 Manfaat

Manfaat yang dapat diperoleh dari pembuatan Tugas Akhir Prototipe Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51 adalah sebagai alat bantu pengemudi mobil saat memparkir mobil.

1.5 Metodologi Penelitian

Dalam pembuatan dan penyusunan tugas akhir ini, dilakukan langkah-langkah sebagai berikut :

1. Perancangan perangkat keras serta perangkat lunak.
2. Pembuatan rangkaian elektronik dan rangkaian mikrokontroler AT89S51.
3. Menguji coba rangkaian yang sudah dibuat.
4. Menganalisa masing-masing rangkaian dan menyimpulkan hasil dari uji coba rangkaian.

1.6 Sistematika Laporan

Sistematika penulisan laporan tugas akhir ini dapat dijelaskan seperti dibawah berikut ini :

1. BAB I PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan dan manfaat, metodologi penelitian dan sistematika penulisan laporan.

2. BAB II LANDASAN TEORI

Berisi teori penunjang yang menguraikan tentang teori-teori yang mendukung dari bagian-bagian perangkat atau alat yang dibuat.

3. BAB III DESAIN DAN PERANCANGAN

Berisi hal-hal yang berhubungan dengan perancangan dan pembahasan perangkat keras tentang alat yang dibuat.

4. BAB IV IMPLEMENTASI DAN ANALISA

Memuat hasil pengamatan dan pembahasan dari hasil pengujian alat yang dibuat.

5. BAB V PENUTUP

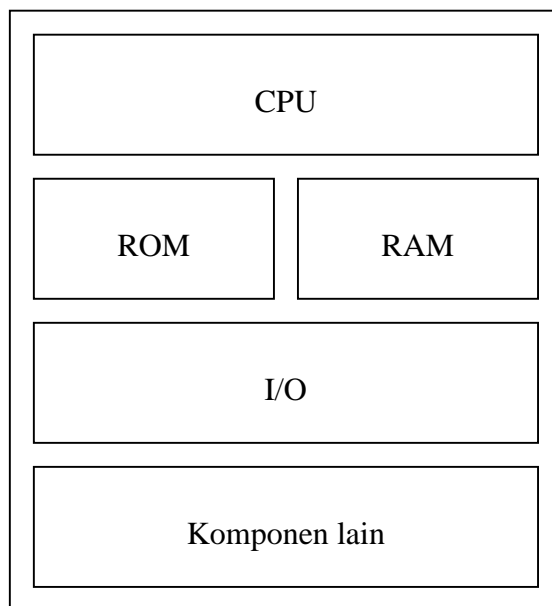
Berisi kesimpulan dan cara tentang penggunaan alat yang telah dirancang sebagai tugas akhir ini.

BAB II

LANDASAN TEORI

2.1 Gambaran Umum Mikrokontroler

Mikrokontroler merupakan suatu IC yang di dalamnya berisi CPU, ROM, RAM, dan I/O. Dengan adanya CPU tersebut maka mikrokontroler dapat melakukan proses berfikir berdasarkan program yang telah diberikan kepadanya. Mikrokontroler banyak terdapat pada peralatan elektronik yang serba otomatis, mesin fax, dan peralatan elektronik lainnya. Mikrokontroler dapat disebut pula sebagai komputer yang berukuran kecil yang berdaya rendah sehingga sebuah baterai dapat memberikan daya. Mikrokontroler terdiri dari beberapa bagian seperti yang terlihat pada gambar di bawah ini (Agfianto Eko Putra, 2004) :



Gambar 2.1
mikrokontroler
Pada
Susunan
gambar tersebut
tampak suatu mikrokontroler standart yang tersusun atas komponen-komponen sebagai berikut :

A. *Central Processing Unit (CPU)*

CPU merupakan bagian utama dalam suatu mikrokontroler. CPU pada mikrokontroler ada yang berukuran 8 bit ada pula yang berukuran 16 bit. CPU ini akan membaca program yang tersimpan di dalam ROM dan melaksanakannya.

B. *Read Only Memory (ROM)*

ROM merupakan suatu memori (alat untuk mengingat) yang sifatnya hanya dibaca saja. Dengan demikian ROM tidak dapat ditulisi. Dalam dunia mikrokontroler ROM digunakan untuk menyimpan program bagi mikrokontroler tersebut. Program tersimpan dalam format biner ('0' atau '1'). Susunan bilangan biner tersebut bila telah terbaca oleh mikrokontroler akan memiliki arti tersendiri.

C. *Random Acces Memory (RAM)*

Berbeda dengan ROM, RAM adalah jenis memori selain dapat dibaca juga dapat ditulis berulang kali. Tentunya dalam pemakaian mikrokontroler ada semacam data yang bisa berubah pada saat mikrokontroler tersebut bekerja. Perubahan data tersebut tentunya juga akan tersimpan ke dalam memori. Isi pada RAM akan hilang jika catu daya listrik hilang.

D. *Input / Output (I/O)*

Untuk berkomunikasi dengan dunia luar, maka mikrokontroler menggunakan terminal I/O (*port I/O*), yang digunakan untuk masukan atau keluaran.

E. *Komponen lainnya*

Beberapa mikrokontroler memiliki *timer/counter*, ADC (*Analog to Digital Converter*), dan komponen lainnya. Pemilihan komponen tambahan yang sesuai dengan tugas mikrokontroler akan sangat membantu perancangan sehingga dapat mempertahankan ukuran yang kecil. Apabila komponen-komponen tersebut belum ada pada suatu mikrokontroler, umumnya komponen tersebut masih dapat ditambahkan pada sistem mikrokontroler melalui port-portnya.

2.2 Mikrokontroler AT89S51

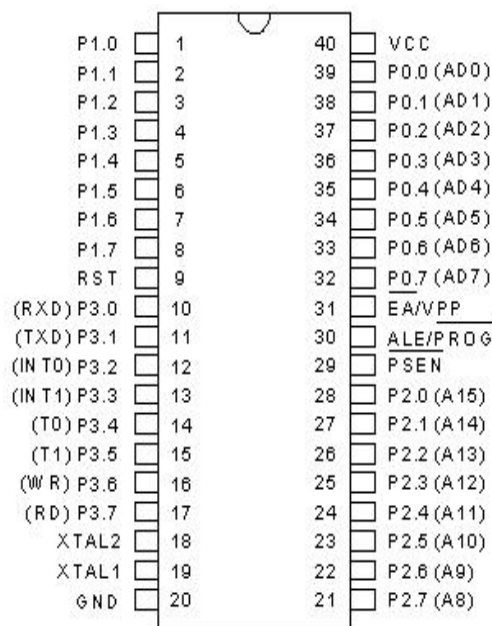
Mikrokontroler AT89S51 adalah mikrokontroler CMOS 8 bit keluaran Atmel dengan kapasitas *Flash memory* sebesar 4K *bytes*. Selain itu AT89S51 juga mempunyai kapasitas RAM sebesar 128 *bytes*, 32 saluran I/O, *Watchdog timer*, dua *pointer data*, dua *timer/counter* 16-bit.

Memori *Flash* digunakan untuk menyimpan perintah (instruksi) berstandar MCS-51, sehingga memungkinkan mikrokontroler ini untuk bekerja dalam mode *single chip operation* (mode operasi keping tunggal) yang tidak memerlukan

external memory (memori luar) untuk menyimpan *source code* tersebut (<http://elektronika21.blogspot.com/2007/11/mikrokontroler-at89s51.html> diakses tanggal 27 Desember 2008).

2.2.1 Arsitektur Mikrokontroler AT89S51

Mikrokontroler ini mempunyai empat port I/O, akumulator, register, RAM internal, *stack pointer*, *Arithmetic Logic Unit (ALU)*, pengunci (*latch*), dan rangkaian osilasi yang membuat mikrokontroler ini dapat beroperasi hanya dengan sekeping IC. Secara fisik, mikrokontroler AT89S51 mempunyai 40 pin, 32 pin diantaranya adalah pin untuk keperluan port masukan atau keluaran. Satu port paralel terdiri dari 8 pin, dengan demikian 32 pin tersebut membentuk 4 buah port paralel, yang masing-masing dikenal dengan Port 0, Port 1, Port 2 dan Port 3. Di bawah ini merupakan susunan pin AT89S51 :



Gambar 2.2 Susunan pin AT89S51

Berikut penjelasan dari masing-masing pin dan port :

1. Port 0

Port 0 merupakan port I/O 8 bit *open drain* dua arah. Sebagai sebuah port, setiap pin dapat mengendalikan 8 input TTL. Ketika logika “1” dituliskan ke port 0, maka port dapat digunakan sebagai input dengan *high* impedansi. Port 0 dapat juga dikonfigurasi untuk multipleksing dengan

address / data bus selama mengakses memori program atau data eksternal. Pada mode ini P0 harus mempunyai *pull up* (<http://mytutorialcafe.com/mikrokontroller/mikrokontrollerdasar.html> diakses tanggal 11 Mei 2009).

2. Port 1

Port 1 merupakan port I/O 8 bit dua arah dengan internal *pull up*. *Buffer* output port 1 dapat mengendalikan empat TTL input. Ketika logika “1” dituliskan ke port 1, maka port ini akan mendapatkan internal *pull up* dan dapat digunakan sebagai input. Port 1 juga menerima alamat *byte* rendah selama pemrograman dan verifikasi *Flash*.

Port Pin Fungsi Alternatif :

P1.5 MOSI (digunakan untuk *In System Programming*)

P1.6 MISO (digunakan untuk *In System Programming*)

P1.7 SCK (digunakan untuk *In System Programming*)

3. Port 2

Port 2 merupakan port I/O 8 bit dua arah dengan internal *pull up*. *Buffer* output port 2 dapat mengendalikan empat TTL input. Ketika logika “1” dituliskan ke port 2, maka port ini akan mendapatkan internal *pull up* dan dapat digunakan sebagai input.

4. Port 3

Port 3 merupakan port I/O 8 bit dua arah dengan internal *pull up*. *Buffer* output port 3 dapat mengendalikan empat TTL input. Ketika logika “1” dituliskan ke port 3, maka port ini akan mendapatkan internal *pull up* dan dapat digunakan sebagai input

(<http://mytutorialcafe.com/mikrokontroller/mikrokontrollerdasar.html> diakses tanggal 11 Mei 2009).

5. Pin 1 sampai 8

Berfungsi sebagai: P1.0 _ P1.7. Pin 1 sampai 8 merupakan saluran I/O 8 bit yang bersifat dua arah, dengan internal *pull-up* yang dapat digunakan untuk berbagai keperluan seperti mengendalikan 4 input TTL. Port ini juga digunakan sebagai saluran alamat saat pemrograman dan verifikasi. Pada pin

6, 7, 8 terdapat port pin yang digunakan pada saat *download* program (<http://elektronika21.blogspot.com/2007/11/mikrokontroler-at89s51.html> diakses tanggal 27 Desember 2008).

6. Pin 9

Berfungsi sebagai : RST. Pin 9 Merupakan masukan reset bagi mikrokontroler. Reset akan aktif dengan memberikan input *high* selama 2 *cycle*.

7. Pin 10 sampai 17

Berfungsi sebagai : P3.0 _ P3.7. Pin 10 sampai 17 merupakan saluran I/O 8 bit dua arah dengan internal *pull-up*. Di samping sebagai saluran I/O, port ini memiliki fungsi pengganti. Bila fungsi pengganti tidak dipakai maka dapat digunakan sebagai port paralel 8 bit serbaguna. Selain itu, sebagian Port 3 dapat berfungsi sebagai sinyal kontrol saat proses pemrograman dan verifikasi.

Tabel 2.1 Fungsi khusus Port 3

Pin	Fungsi
P3.0	RXD masukan port serial
P3.1	TXD keluaran port serial
P3.2	INT0 masukan interupsi 0
P3.3	INT1 masukan interupsi 1
P3.4	T0 masukan <i>Timer/Counter</i> 0
P3.5	T1 masukan <i>Timer/Counter</i> 1
P3.6	WR pulsa penulisan data memori luar
P3.7	RD pulsa pembacaan data memori luar

8. Pin 18

Berfungsi sebagai : XTAL2. Pin 18 merupakan keluaran dari rangkaian osilasi mikrokontroler.

9. Pin 19

Berfungsi sebagai : XTAL1. Pin 19 merupakan masukan untuk rangkaian osilasi mikrokontroler.

10. Pin 20

Berfungsi sebagai : GND. Pin 20 merupakan *ground* dari sumber tegangan.

11. Pin 21 sampai 28

Berfungsi sebagai : P2.0 _ P2.7. Pin 21 sampai 28 merupakan saluran I/O 8 bit dua arah dengan internal *pull-up*. Saat pengambilan data dari program memori eksternal atau selama pengaksesan data memori eksternal yang menggunakan alamat 16 bit. Port 2 berfungsi sebagai saluran alamat tinggi (A8 – A15). Akan tetapi, saat mengakses data memori eksternal yang menggunakan alamat 8 bit, Port 2 mengeluarkan isi P2 pada *Special Function Register*.

12. Pin 29

Berfungsi sebagai : PSEN. Pin ini berfungsi pada saat mengeksekusi program yang terletak pada memori eksternal. *Program Strobe Enable* (PSEN) akan aktif dua kali setiap *cycle*.

13. Pin 30

Berfungsi sebagai : ALE/PROG. Pin ini dapat berfungsi sebagai *Address Latch Enable* (ALE) yang menahan *low bytes address* pada saat mengakses memori eksternal. Sedangkan pada saat *Flash Programming* (PROG) berfungsi sebagai pulsa input selama proses pemrograman.

14. Pin 31

Berfungsi sebagai : EA/VPP. Pada kondisi *low*, pin ini akan berfungsi sebagai *External Access Enable* (EA) yaitu mikrokontroler akan menjalankan program yang ada pada memori eksternal. Jika berkondisi *high*, pin ini akan berfungsi untuk menjalankan program yang ada pada memori internal. Pin ini juga berfungsi sebagai masukan tegangan selama proses pemrograman.

15. Pin 32 sampai 39

Berfungsi sebagai : D7 _ D0 (A7 _A0). Pin 32 sampai 39 ialah Port 0 yang merupakan saluran I/O 8 bit *open collector* dan dapat juga digunakan sebagai multipleks bus alamat rendah dan bus data selama adanya akses ke memori program eksternal. Saat proses pemrograman dan verifikasi, Port 0 digunakan sebagai saluran data.

16. Pin 40

Berfungsi sebagai : VCC. Pin 40 merupakan masukan sumber tegangan positif bagi mikrokontroler.
(<http://elektronika21.blogspot.com/2007/11/mikrokontroler-at89s51.html> diakses tanggal 27 Desember 2008).

2.2.2 Memori Internal AT89S51

Memori internal AT89S51 terdiri dari 3 bagian yaitu ROM, RAM dan SFR.

1. *Read Only Memory* (ROM)

ROM adalah memori tempat menyimpan program / *source code*. Sifat ROM adalah *non-volatile* yaitu data / program tidak akan hilang walaupun tegangan *supply* tidak ada. Kapasitas ROM tergantung dari tipe mikrokontroler. Untuk AT89S51 kapasitas ROM adalah 4 KByte. ROM pada AT89S51 menempati *address* 0000 s/d 0FFF (<http://www.polibatam.ac.id> diakses tanggal 27 Desember 2008).

2. *Random Access Memory* (RAM)

RAM adalah memori tempat menyimpan data sementara. Sifat RAM adalah *volatile* yaitu data akan hilang jika tegangan *supply* tidak ada. Kapasitas RAM tergantung pada tipe mikrokontroler. Pada AT89S51 RAM dibagi menjadi 2 yaitu :

A. *LOWER 128 byte yang menempati address 00 s/d 7F.*

RAM ini dapat diakses dengan pengalamatan langsung (*direct*) maupun tak langsung (*indirect*). Contoh :

Direct => `mov 30h,#120` ; Pindahkan data 120 ke RAM pada *address* 30h

Indirect => `mov R0,#30h` ; Isi Register 0 dengan 30h

mov @R0,#120 ; Pindahkan data 120 ke RAM pada *address*
; sesuai isi R0

B. UPPER 128 byte yang menempati *address* 80 s/d FF.

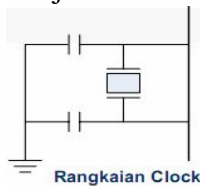
Address ini sama dengan *address* SFR meski secara fisik benar – benar berbeda. RAM ini hanya dapat diakses dengan pengalamatan tak langsung saja.

3. Special Function Register (SFR)

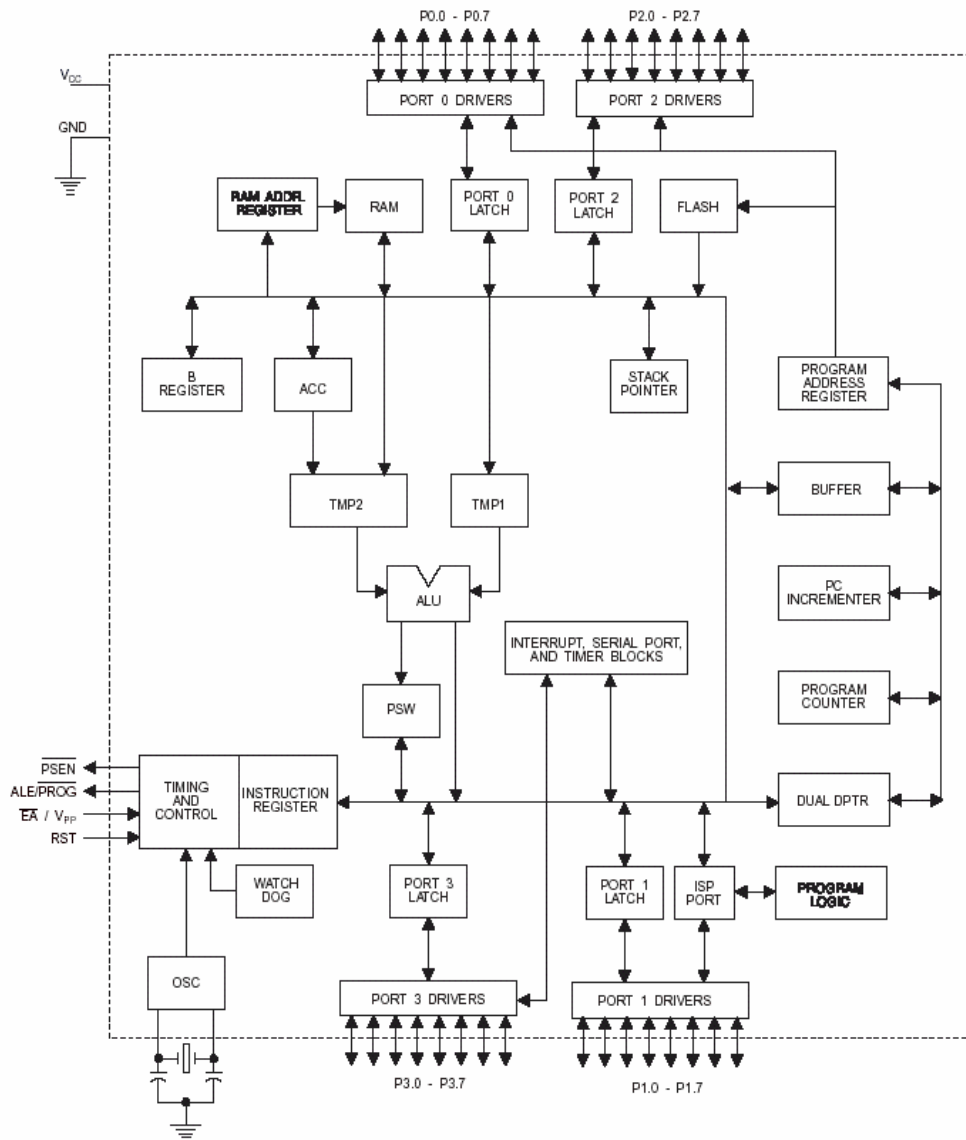
SFR adalah register dengan fungsi tertentu. Misalnya, register TMOD dan TCON adalah *timer control* register yang berfungsi mengatur fasilitas *timer* mikrokontroler. SFR pada AT89S51 menempati *address* 80 s/d FF.

2.2.3 Osilator dan Clock

Agar dapat mengeksekusi program, mikrokontroler membutuhkan pulsa *clock*. Pulsa ini dapat dihasilkan dengan memasang rangkaian resonator pada pin XTAL1 dan XTAL2. Frekuensi kerja maksimum AT89S51 adalah 33 MHz. Gambar di bawah adalah contoh rangkaian osilator yang bisa digunakan pada mikrokontroler. Komponen utamanya adalah *quartz crystal* yang dihubungkan dengan kapasitor. Nilai kapasitornya biasanya 33pF (<http://www.polibatam.ac.id> diakses tanggal 27 Desember 2008).



Gambar 2.3 Rangkaian *clock*



Gambar 2.4 Diagram blok AT89S51

2.2.4 Bahasa Assembly Mikrokontroler AT89S51

Secara fisik, mikrokontroler bekerja dengan membaca instruksi yang tersimpan di dalam memori. Mikrokontroler menentukan alamat dari memori program yang akan dibaca dan melakukan proses baca data di memori. Data yang dibaca diinterpretasikan sebagai instruksi. Alamat instruksi disimpan oleh mikrokontroler di register, yang dikenal sebagai program *counter*. Instruksi ini misalnya program aritmatika yang melibatkan 2 register (<http://www.toko-elektronika.com/tutorial/uc2.html> diakses tanggal 27 Desember 2008).

Mikrokontroler AT89S51 memiliki sekumpulan instruksi yang sangat

lengkap. Instruksi MOV untuk *byte* dan bit dikelompokkan sesuai dengan mode pengalamatan (*addressing modes*). Mode pengalamatan menjelaskan bagaimana *operand* dioperasikan. Bentuk program *assembly* yang umum ialah sebagai berikut :

Label	<i>mnemonic</i>	<i>operand 1</i>	<i>operand 2</i>	komentar
(isi memori)	(<i>opcode</i>)			
↓	↓	↓	↓	↓
4000 7430	MOV	A, #35H		;copy 35H ke akumulator A

Isi memori ialah bilangan heksadesimal yang dikenal oleh mikrokontroler yang merupakan representasi dari bahasa *assembly* yang telah dibuat. ***Mnemonic*** atau ***opcode*** ialah kode yang akan melakukan aksi terhadap *operand*. ***Operand*** ialah data yang diproses oleh *opcode*. Sebuah *opcode* bisa membutuhkan 1, 2 atau lebih *operand*, kadang juga tidak perlu *operand*. Sedangkan **komentar** dapat menggunakan tanda titik koma (;). Berikut contoh jumlah *operand* yang berbeda-beda dalam suatu *assembly* (<http://www.toko-elektronika.com/tutorial/uc2.html> diakses tanggal 27 Desember 2008).

CJNE R5,#22H, aksi	;dibutuhkan 3 buah <i>operand</i>
MOVX @DPTR, A	;dibutuhkan 2 buah <i>operand</i>
RL A	;1 buah <i>operand</i>
NOP	;tidak memerlukan <i>operand</i>

2.2.5 Instruksi Mikrokontroler AT89S51

Instruksi pada mikrokontroler digunakan untuk menjalankan program sesuai dengan perintah yang diinginkan. Di bawah ini merupakan instruksi yang dapat digunakan untuk memprogram mikrokontroler AT89S51 (<http://www.toko-elektronika.com/tutorial/lampiran.html> diakses tanggal 7 Januari 2009).

1. ACALL (*Absolute Call*)

Instruksi ACALL digunakan untuk memanggil sub rutin program (<http://www.toko-elektronika.com/tutorial/lampiran.html> diakses tanggal 7

Januari 2009).

Contoh :

START:

ACALL TUNDA ; Panggil *Procedure* penundaan waktu

....

TUNDA: ; Label Tunda

MOV R7,#0FFH ; Isikan Register 7 dengan data 0FFH(255)

2. ADD (Add Immediate Data)

Instruksi ini akan menambah 8 bit data langsung ke dalam isi akumulator dan menyimpan hasilnya pada akumulator.

Contoh : Add A, #data

Add A, #@R1 ; Add indirect address

Add A, R6 ; Add register

Add A, 30H ; Add memori

3. CJNE (Compare Indirect Address to Immediate Data)

Instruksi ini akan membandingkan data langsung dengan lokasi memori yang dialamati oleh register R atau Akumulator A. Apabila tidak sama maka instruksi akan menuju ke alamat kode.

Format : CJNE R,#data,Alamat kode

Contoh:

CJNE R7,#001H,Command()

MOV A,StepControl

AJMP Command1

4. CLR (Clear Accumulator)

Instruksi CLR akan mereset data akumulator menjadi 00H.

Format : CLR A

5. DEC (Decrement Indirect Address)

Instruksi DEC akan mengurangi isi lokasi memori yang ditujukan oleh register R dengan 1 dan hasilnya disimpan pada lokasi tersebut.

Contoh: DEC 40H

DEC R7 ; decrement register

6. DJNZ (*Decrement Register And Jump If Not Zero*)

Instruksi DJNZ akan mengurangi nilai register dengan 1 dan jika hasilnya sudah 0 maka instruksi selanjutnya akan dieksekusi. Jika belum 0 akan menuju ke alamat kode.

Format : DJNZ Rr,Alamat Kode

7. INC (*Increment Indirect Address*)

Instruksi INC akan menambahkan isi memori dengan 1 dan menyimpannya pada alamat tersebut.

Contoh: INC A

INC R7 ; *increment register*

8. JMP (*Jump to sum of Accumulator and Data Pointer*)

Instruksi JMP untuk memerintahkan loncat kesuatu alamat kode tertentu.

Format : JMP alamat kode.

Contoh :

Loop:

...

RL A ; Geser data Akumulator ke kiri

ACALL Long_Delay ; Panggil *Procedure* penundaan waktu

JMP Loop ; Loncat ke *Procedure Loop*

9. MOV

Instruksi ini untuk memindahkan isi akumulator/register atau data dari nilai luar atau alamat lain.

Contoh :

MOV A,#40H

MOV @RO,A

MOV C, P1.0

MOV DPTR, #20H

MOVC A, @A+DPTR ; pindahkan kode memori *offset* dari data *pointer* ke A

MOVX @DPTR, A ; Pindahkan akumulator ke memori eksternal yang dialamati
; oleh *data pointer*

10. RET (*Return from subroutine*)

Instruksi untuk kembali dari suatu subrutin program ke alamat terakhir subrutin tersebut di panggil.

11. SETB (Set Bit)

Instruksi SETB untuk mengaktifkan atau memberikan logika 1 pada sebuah bit data.

Format :

SETB A.1 (memberikan logika 1 pada *accumulator* bit ke 1)

SETB P1.1 (memberikan logika 1 pada Port 1 bit ke 1)

12. CLRB (Clear Bit)

Instruksi CLRB untuk memberikan logika 0 pada sebuah bit data.

Format :

CLRB A.1 ; memberikan logika 0 pada *accumulator* bit ke 1

CLRB P1.1 ; memberikan logika 0 pada Port 1 bit ke 1

2.3 Sensor

Sensor adalah alat untuk mendeteksi / mengukur sesuatu yang digunakan untuk mengubah variasi mekanis, magnetis, panas, sinar dan kimia menjadi tegangan dan arus listrik. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian akan diolah oleh kontroler sebagai otaknya. Sensor dalam teknik pengukuran dan pengaturan secara elektronik berfungsi mengubah besaran fisik (misalnya : temperatur, gaya, kecepatan putaran) menjadi besaran listrik yang proposional (<http://indomicron.co.cc/elektronika/analog/pengertian-sensor/> diakses tanggal 25 Mei 2009).

Salah satu sensor yang digunakan dalam pembuatan tugas akhir ini adalah sensor ultrasonik. Penjelasan dari sensor ultrasonik terdapat di bawah ini.

2.3.1 Sensor Ultrasonik

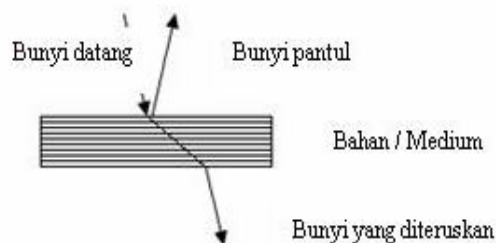
A. Pengertian Gelombang Ultrasonik

Gelombang ultrasonik merupakan gelombang akustik yang memiliki frekuensi mulai 20 kHz hingga sekitar 20 MHz. Frekuensi kerja yang digunakan dalam gelombang ultrasonik bervariasi tergantung pada medium yang dilalui, mulai dari kerapatan rendah pada fasa gas, cair hingga padat. Jika gelombang

ultrasonik berjalan melalui sebuah medium, secara matematis besarnya jarak dapat dihitung sebagai berikut : $s = v.t/2$ (<http://indomicron.co.cc/> diakses tanggal 21 April 2009).

Dimana s adalah jarak dalam satuan meter, v adalah kecepatan suara yaitu 344 m/detik dan t adalah waktu tempuh dalam satuan detik. Ketika gelombang ultrasonik menumbuk suatu penghalang maka sebagian gelombang tersebut akan dipantulkan sebagian diserap dan sebagian yang lain akan diteruskan (<http://indomicron.co.cc/> diakses tanggal 21 April 2009).

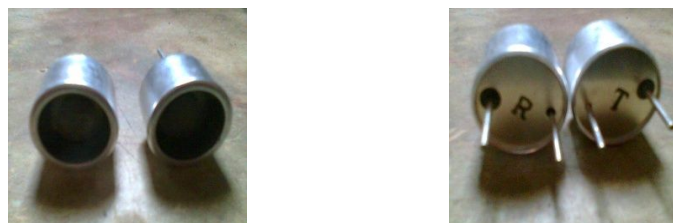
Proses ini ditunjukkan pada gambar berikut :



Gambar 2.5 Proses pemantulan gelombang ultrasonik

B. Pengertian Sensor Ultrasonik

Sensor ultrasonik adalah sebuah sensor yang mengubah besaran fisis (bunyi) menjadi besaran listrik. Pada sensor ini gelombang ultrasonik dibangkitkan melalui sebuah benda yang disebut *piezoelektrik*. *Piezoelektrik* ini akan menghasilkan gelombang ultrasonik dengan frekuensi 40 kHz ketika sebuah osilator diterapkan pada benda tersebut (<http://indomicron.co.cc/> diakses tanggal 21 April 2009).



Gambar 2.6 Bentuk fisik sensor ultrasonik model biasa

Sensor ultrasonik secara umum digunakan untuk suatu pengungkapan tak sentuh yang beragam seperti aplikasi pengukuran jarak. Alat ini secara umum memancarkan gelombang suara ultrasonik menuju suatu target yang

memantulkan balik gelombang ke arah sensor. Kemudian sistem mengukur waktu yang diperlukan untuk pemancaran gelombang sampai kembali ke sensor dan menghitung jarak target dengan menggunakan kecepatan suara dalam medium (<http://indomicron.co.cc/> diakses tanggal 21 April 2009).

Rangkaian penyusun sensor ultrasonik ini terdiri dari *transmitter*, *receiver*, dan komparator. Selain itu, gelombang ultrasonik dibangkitkan oleh sebuah kristal tipis bersifat *piezoelektrik*. Bagian-bagian dari sensor ultrasonik adalah sebagai berikut :

1. *Piezoelektrik*

Peralatan *piezoelektrik* secara langsung mengubah energi listrik menjadi energi mekanik. Tegangan input yang digunakan menyebabkan bagian keramik meregang dan memancarkan gelombang ultrasonik. Tipe operasi transmisi elemen *piezoelektrik* sekitar frekuensi 32 kHz. Efisiensi lebih baik, jika frekuensi osilator diatur pada frekuensi resonansi piezoelektrik dengan sensitifitas dan efisiensi paling baik. Jika rangkaian pengukur beroperasi pada mode pulsa elemen piezoelektrik yang sama dapat digunakan sebagai *transmitter* dan *receiver* (<http://indomicron.co.cc/> diakses tanggal 21 April 2009).

2. *Transmitter*

Transmitter adalah sebuah alat yang berfungsi sebagai pemancar gelombang ultrasonik dengan frekuensi sebesar 40 kHz yang dibangkitkan dari sebuah osilator. Untuk menghasilkan frekuensi 40 KHz, harus dibuat sebuah rangkaian osilator dan keluaran dari osilator dilanjutkan menuju penguat sinyal. Besarnya frekuensi ditentukan oleh komponen kalang RLC / kristal tergantung dari desain osilator yang digunakan. Penguat sinyal akan memberikan sebuah sinyal listrik yang diumpankan ke piezoelektrik dan terjadi reaksi mekanik sehingga bergetar dan memancarkan gelombang yang sesuai dengan besar frekuensi pada osilator (<http://indomicron.co.cc/> diakses tanggal 21 April 2009).

3. *Receiver*

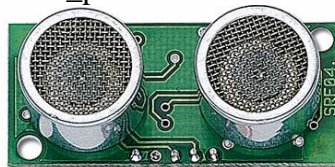
Receiver terdiri dari transduser ultrasonik menggunakan bahan *piezoelektrik*, yang berfungsi sebagai penerima gelombang pantulan yang berasal dari *transmitter* yang dikenakan pada permukaan suatu benda atau gelombang

langsung LOS (*Line of Sight*) dari *transmitter*. Oleh karena bahan *piezoelektrik* memiliki reaksi yang *reversible*, elemen keramik akan membangkitkan tegangan listrik pada saat gelombang datang dengan frekuensi yang resonan dan akan menggetarkan bahan *piezoelektrik* tersebut (<http://indomicron.co.cc/> diakses tanggal 21 April 2009).

C. Sensor Jarak Ultrasonik Devantech SRF04

SRF04 adalah sensor non-kontak pengukur jarak menggunakan ultrasonik. Prinsip kerja sensor ini adalah *transmitter* mengirimkan seberkas gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek. Lamanya waktu ini sebanding dengan dua kali jarak sensor dengan obyek, sehingga jarak sensor dengan obyek dapat ditentukan persamaan :

$\text{jarak} = \text{kecepatan_suara} \times \text{waktu_pantul} / 2.$



Gambar 2.7 Bentuk fisik SRF04

Spesifikasi teknis Devantech SRF04 Ultrasonic Range Finder:

1. Tegangan : 5 VDC
2. Konsumsi Arus : 30 mA (rata-rata), 50 mA (max)
3. Frekuensi Suara : 40 kHz
4. Jangkauan : 3 cm - 3 m
5. Sensitivitas : Mampu mendeteksi gagang sapu berdiameter 3 cm dalam jarak > 2 m
6. Input Trigger : 10 mS min. Pulsa Level TTL
7. Pulsa Echo : Sinyal level TTL positif, lebar berbanding proporsional dengan jarak yang dideteksi

SRF04 dapat mengukur jarak dalam rentang antara 3 cm – 3 m dengan *output* panjang pulsa yang sebanding dengan jarak obyek. Sensor ini hanya memerlukan 2 pin I/O untuk berkomunikasi dengan mikrokontroler, yaitu TRIGGER dan ECHO. Untuk mengaktifkan SRF04 mikrokontroler mengirimkan pulsa positif melalui pin TRIGGER minimal 10 μ s, selanjutnya SRF04 akan mengirimkan pulsa positif melalui pin ECHO selama 100 μ s hingga 18 ms, yang sebanding dengan jarak obyek. Dibandingkan dengan sensor ultrasonik lain, seperti PING, SRF04 mempunyai kemampuan yang setara, yaitu rentang pengukuran antara 3 cm – 3 m, dan output yang sama, yaitu panjang pulsa. Meski

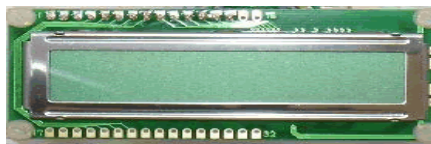
cara pengoperasiannya juga mirip, namun kedua sensor tersebut berbeda jumlah pin I/O-nya, yaitu 2 untuk SRF04 dan 1 untuk PING

(<http://roboticunpad.blogspot.com/2009/04/sensor.html> diakses tanggal 7 Juli 2009).

2.4 *Liquid Crystal Display (LCD)*

LCD merupakan salah satu komponen penting dalam pembuatan tugas akhir ini karena LCD dapat menampilkan perintah-perintah yang harus dijalankan oleh pemakai. LCD mempunyai kemampuan untuk menampilkan tidak hanya angka, huruf abjad, kata-kata tapi juga simbol-simbol.

Jenis dan ukuran LCD bermacam-macam, antara lain 2x16, 2x20, 2x40, dan lain-lain. LCD mempunyai dua bagian penting yaitu *backlight* yang berguna jika digunakan pada malam hari dan *contrast* yang berfungsi untuk mempertajam tampilan (<http://digilib.petra.ac.id/prototype-chapter3.pdf> diakses tanggal 21 April 2009).



Gambar 2.8 Bentuk fisik LCD 2x16 karakter

Tabel 2.2 Fungsi pin LCD

No	Nama Pin	Fungsi
1	VSS	GND

2	VDD	Suplai tegangan +5V
3	VLC	Tegangan kontras LCD
4	RS	L = input instruksi, H = input data
5	R/W	L = tulis data dari MPU ke LCM, H = baca data dari LCM ke MPU
6	E	<i>Enable Clock</i>
7	DB0	<i>Data Bus Line</i>
8	DB1	<i>Data Bus Line</i>
9	DB2	<i>Data Bus Line</i>
10	DB3	<i>Data Bus Line</i>
11	DB4	<i>Data Bus Line</i>
12	DB5	<i>Data Bus Line</i>
13	DB6	<i>Data Bus Line</i>
14	DB7	<i>Data Bus Line</i>
15	Anoda	Tegangan positif <i>backlight</i>
16	Katoda	Tegangan negatif <i>backlight</i>

Fungsi dari masing – masing pin pada LCD adalah pin pertama dan kedua merupakan pin untuk tegangan suplai sebesar 5 volt, untuk pin ketiga harus ditambahkan resistor variabel 4K7 atau 5K ke pin ini sebagai pengatur kontras tampilan yang diinginkan.

Pin keempat berfungsi untuk memasukkan *input command* atau input data, jika ingin memasukkan *input command* maka pin 4 diberikan *logic low* (0), dan jika ingin memasukkan input data maka pin 4 diberikan *logic high* (1).

Fungsi pin kelima untuk *read* atau *write*, jika diinginkan untuk membaca karakter data atau status informasi dari register (*read*) maka harus diberi masukan *high* (1), begitu pula sebaliknya untuk menuliskan karakter data (*write*) maka harus diberi masukan *low* (0). Pada pin ini dapat dihubungkan ke *ground* bila tidak diinginkan pembacaan dari LCD dan hanya dapat digunakan untuk mentransfer data ke LCD.

Pin keenam berfungsi sebagai *enable*, yaitu sebagai pengatur transfer *command* atau karakter data ke dalam LCD. Untuk menulis ke dalam LCD data ditransfer waktu terjadi perubahan dari *high* ke *low*, untuk membaca dari LCD dapat dilakukan ketika terjadi transisi perubahan dari *low* ke *high*.

Pin-pin dari nomor 7 sampai 14 merupakan data 8 bit yang dapat ditransfer dalam 2 bentuk yaitu 1 kali 8 bit atau 2 kali 4 bit, pin-pin ini akan langsung terhubung ke pin-pin mikrokontroler sebagai input/output. Untuk pin nomor 15-16 berfungsi sebagai *backlight* (<http://digilib.petra.ac.id/prototype-chapter3.pdf> diakses tanggal 21 April 2009).

2.5 Loudspeaker

Loudspeaker atau sering disebut pengeras suara adalah komponen elektronika yang mampu mengubah sinyal listrik menjadi sinyal suara. Proses mengubah sinyal ini dilakukan dengan cara menggerakkan komponennya yang berbentuk selaput (Prihono,dkk , 2009:30).



Gambar 2.9 Bentuk fisik *loudspeaker*

Dalam setiap sistem penghasil suara, penentuan kualitas suara terbaik tergantung dari *speaker*. Pada dasarnya, *speaker* merupakan mesin penerjemah

akhir, kebalikan dari mikrofon. *Speaker* membawa sinyal elektrik dan mengubahnya kembali menjadi getaran untuk membuat gelombang suara. *Speaker* menghasilkan getaran yang hampir sama dengan yang dihasilkan oleh mikrofon yang direkam dan dikodekan (Prihono,dkk , 2009:30).

BAB III

DESAIN DAN PERANCANGAN

3.1 Analisis Kebutuhan

Dalam pembuatan prototipe sensor parkir mobil ini membutuhkan beberapa perangkat *hardware* dan *software* antara lain :

3.1.1 Hardware

1. Minimum Sistem AT89S51

Rangkaian ini bisa disebut sebagai *CPU board* yang berfungsi sebagai pengendali utama dari keseluruhan sistem atau dapat disebut sebagai otak. Rangkaian ini dilengkapi dengan port-port dimana *CPU board* dapat berhubungan dengan modul-modul pendukung yang lain. Minimum sistem AT89S51 menggunakan *chip* AT89S51.

2. Sensor Ultrasonik SRF04

Rangkaian sensor ultrasonik SRF04 terdiri dari TX (*transmitter*) dan RX (*receiver*). TX berfungsi sebagai pemancar sinyal yang mengenai penghalang sedangkan RX berfungsi sebagai penerima sinyal pantulan dari TX.

3. Display LCD

Perangkat ini digunakan sebagai output atau penampil dari hasil yang sudah diproses pada mikrokontroler.

4. Speaker

Speaker digunakan sebagai indikator bunyi pada sensor parkir. Suara yang dihasilkan oleh *speaker* di seting bervariasi sesuai dengan jarak parkir. Jika jarak parkir semakin dekat dengan obyek penghalang di depan sensor, maka bunyi dari *speaker* akan semakin keras.

3.1.2 *Software*

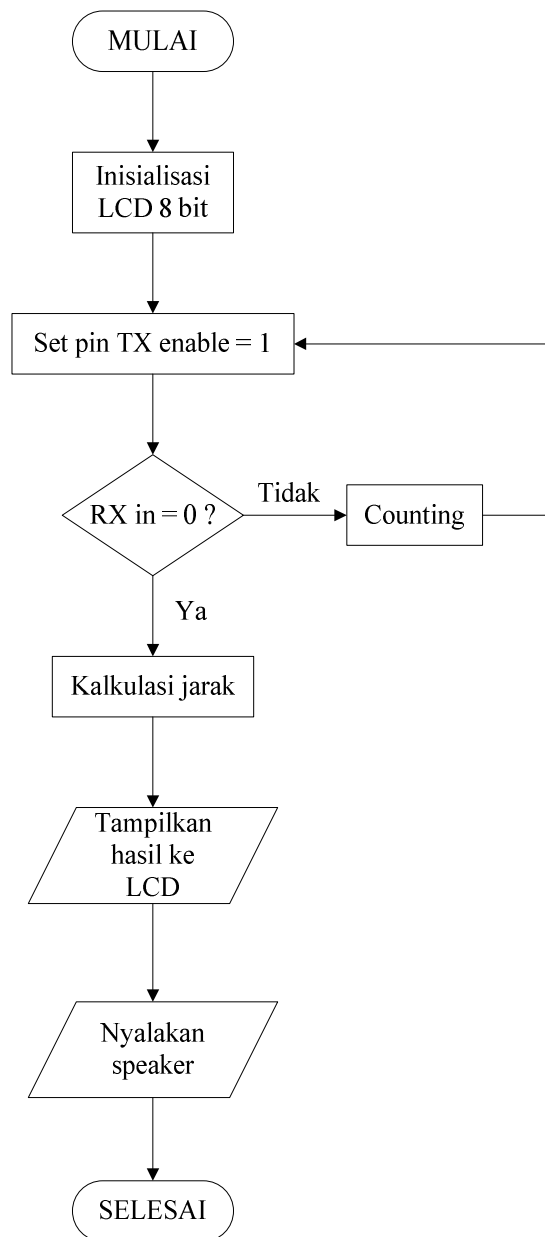
1. **Proteus 7 Profesional**

Proteus sebagai program yang digunakan untuk merancang rangkaian elektronik.

2. **Program *compiler* ASM51 dan program *downloader* AEC ISP**

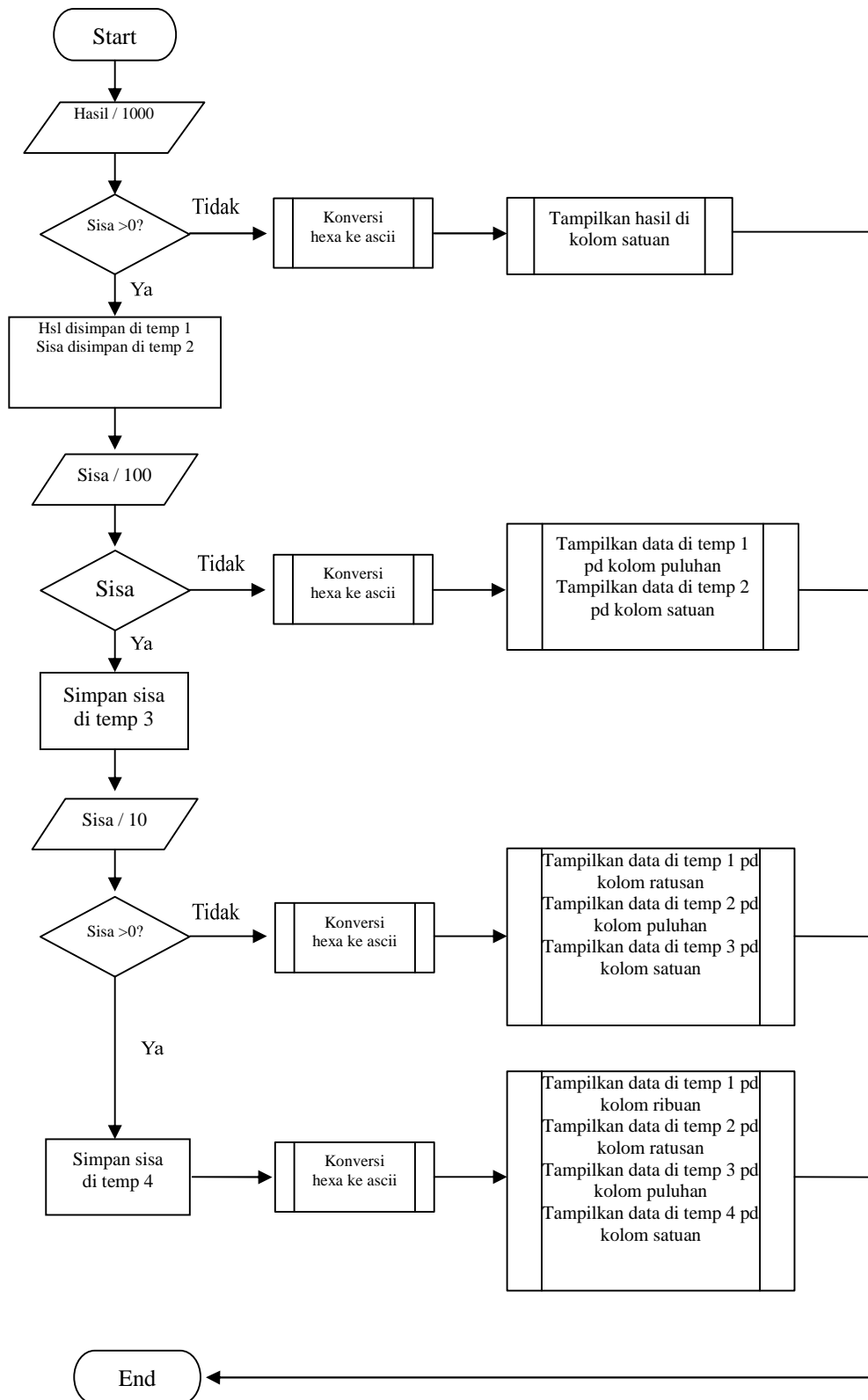
ASM51 adalah program *compiler* berbasis *windows* untuk mikrokontroler keluarga ATMEL. Pemrograman pada mikrokontroler AT89S51 menggunakan bahasa tingkat tinggi yaitu bahasa *Assembler*. Fungsi dari program *compiler* ASM51 adalah untuk me-load file berekstensi “.asm” yang sudah dibuat dengan menggunakan *Notepad* untuk dirubah menjadi file berekstensi “.hex”. Setelah file dirubah menjadi “.hex” kemudian di-load dengan menggunakan program *compiler* AEC ISP. Tujuannya adalah untuk memasukkan program mikro ke dalam *downloader* mikrokontroler AT89S51.

3.2 Diagram Alir Sensor Parkir Mobil



Gambar 3.1 *Flowchart* cara kerja sensor parkir

3.2.1 Diagram Alir Kalkulasi Data (Konversi HEXA → DECIMAL)



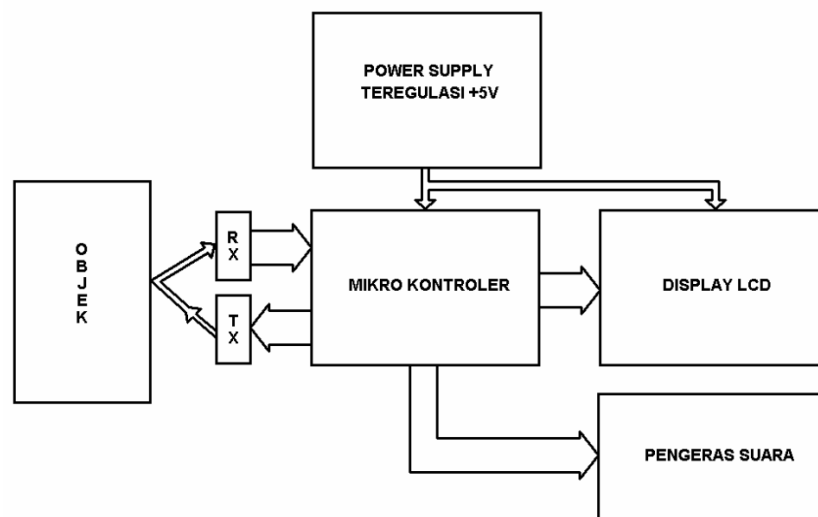
Gambar 3.2 Flowchart kalkulasi data

3.3 Rancangan Skematik

Perancangan papan rangkaian menggunakan *software* Proteus 7 Profesional. Langkah pertama adalah menggambar skema rangkaian pada *schematic editor*. Kemudian dari *schematic editor* komponen yang dirangkai dipindahkan ke *layout* PCB.

1. Diagram Blok Prototipe Sensor Parkir Mobil

Perancangan diagram blok ini dimaksudkan untuk mempermudah pembuatan alat prototipe sensor parkir mobil.



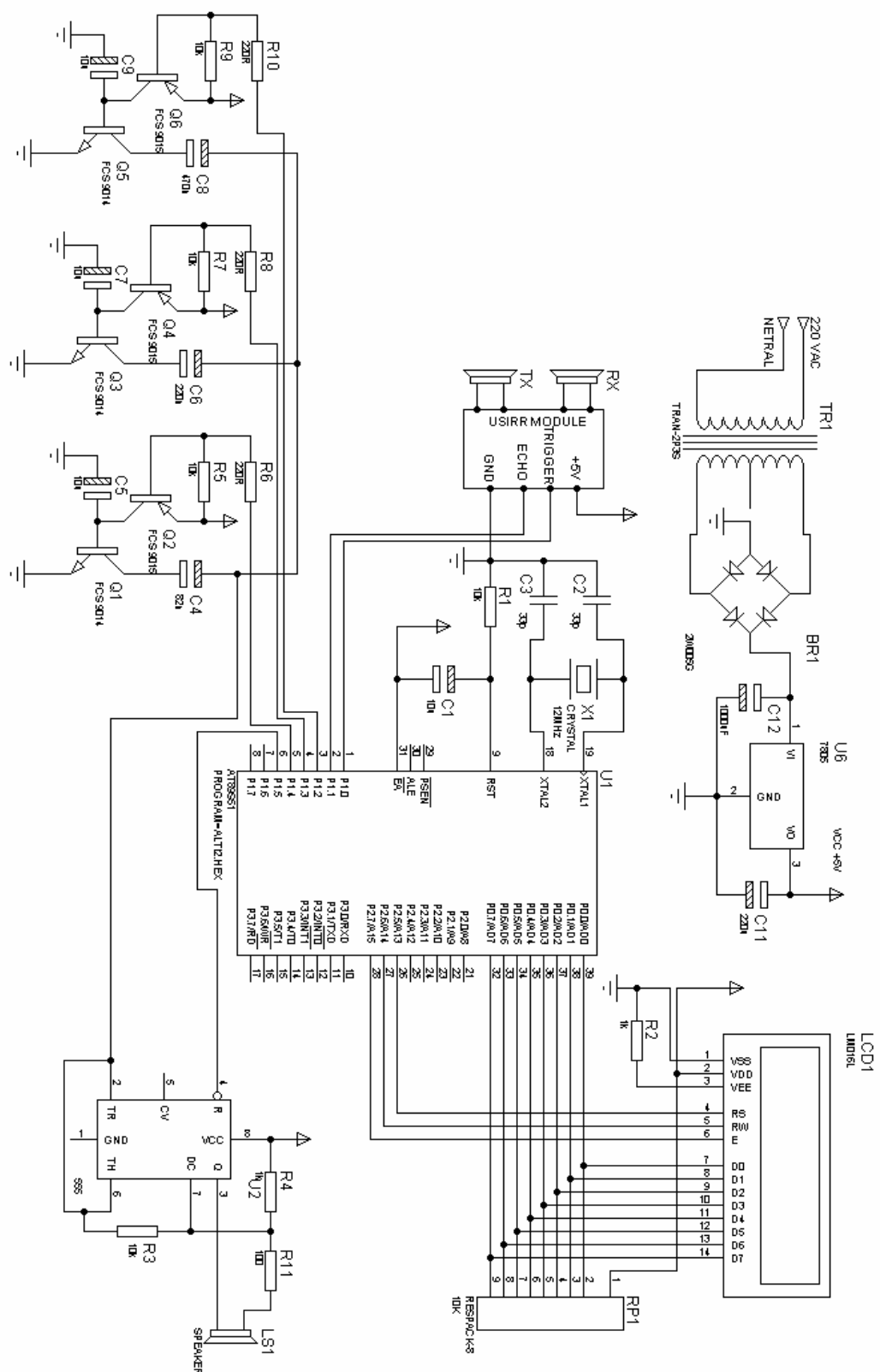
Gambar 3.3 Diagram blok prototipe sensor parkir mobil

2. Desain Rangkaian Prototipe Sensor Parkir Mobil

Dalam pembuatan rangkaian ini dibutuhkan beberapa komponen pokok yaitu sensor ultrasonik SRF04 yang terdiri dari pengirim (*transmitter*) dan penerima (*receiver*), mikrokontroler AT89S51, LCD untuk menampilkan jarak parkir dan *speaker* sebagai indikator bunyi.

Transmitter (TX) mengirimkan sebuah sinyal jika mengenai penghalang kemudian sinyal pantulan akan diterima *Receiver* (RX) yang kemudian dikirimkan ke mikrokontroler yang didalamnya terdapat program untuk menghitung jarak dan diperlihatkan hasilnya melalui LCD. Sebagai indikator suara, rangkaian ini dilengkapi dengan *speaker*. Bunyi dari *speaker*

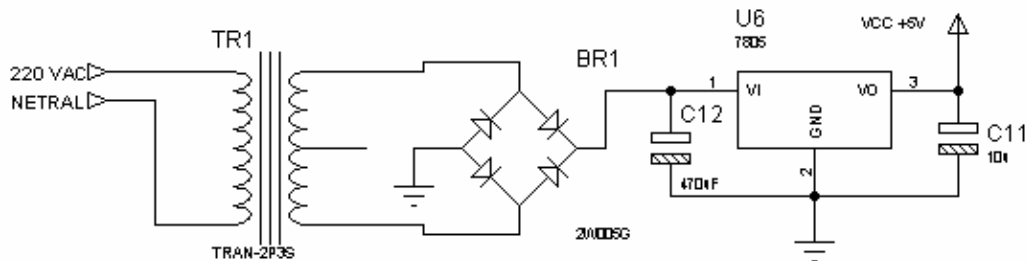
diseting sesuai dengan jarak antara sensor ultrasonik SRF04 dengan penghalang. Jika jarak parkir terlalu dekat dengan penghalang, maka bunyi *speaker* akan semakin tinggi.



Gambar 3.4 Desain rangkaian prototipe sensor parkir mobil

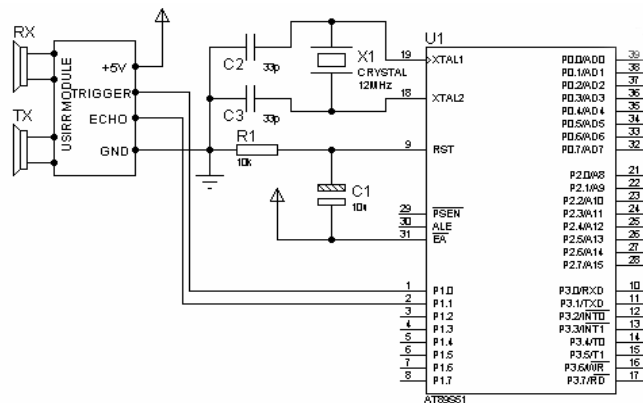
3. Rangkaian Catu daya

Sumber tegangan yang digunakan adalah sebesar 5V/500 mA DC dengan menggunakan trafo regulator.



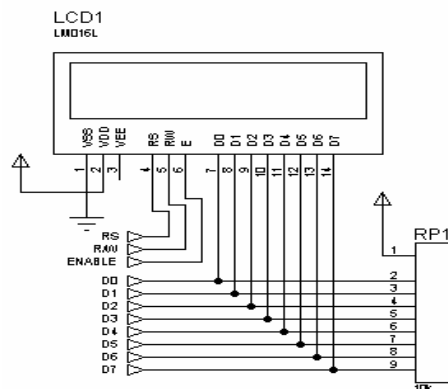
Gambar 3.5 Rangkaian regulator 5V/500mA DC

4. Rangkaian Modul Sensor Ultrasonik SRF04



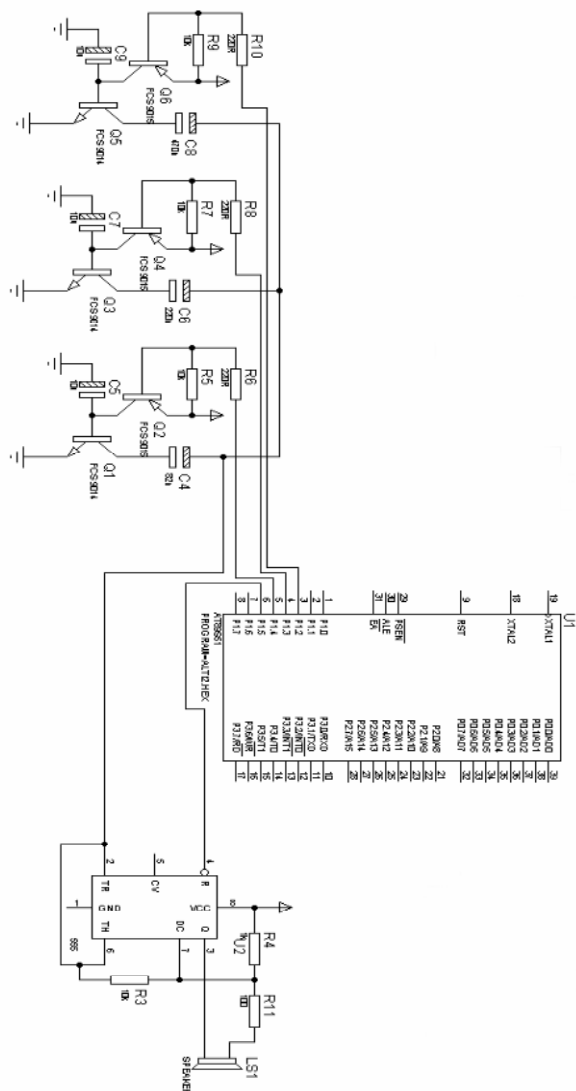
Gambar 3.6 Rangkaian sensor ultrasonik

5. Rangkaian Display



Gambar 3.7 Rangkaian display LCD

6. Rangkaian *Speaker*



Gambar 3.8 Rangkaian *speaker*

3.3.1 Mencetak PCB

Pola dan gambar jalur yang telah dibuat melalui Proteus 7 Profesional kemudian dicetak ke dalam *board* melalui tahapan sebagai berikut :

1. Mencetak gambar *layout* PCB yang telah dibuat pada kertas transparan.
2. Proses berikutnya adalah penyablonan secara langsung diatas lembaran PCB dengan menggunakan sekrin. Sekrin yang digunakan harus sudah terbentuk gambar *layout* PCB pada permukaan sekrin tersebut.

3. Memasukkan yang telah tersablon ke dalam air hangat yang telah dilarutkan bubuk Ferri Clorite. Wadah yang digunakan harus selain logam.
4. Rendam PCB ke dalam larutan Ferri Clorite selama 5-10 menit dengan perbandingan 10 gram bubuk Ferri Clorite untuk 100cc air panas. Goyang-goyang wadah atau tempat perendam PCB agar seluruh lapisan tembaga yang tidak tertutup pola jalur PCB dari sablon dapat terkikis habis lebih cepat.
5. PCB dibersihkan dengan menggunakan kapas untuk menghilangkan sisa-sisa larutan Ferri Clorite dari papan PCB. Untuk menghilangkan sablon menggunakan tiner/bensin.
6. Proses pelubangan PCB menggunakan bor PCB dengan diameter bor 0,8 mm atau 1,0 mm.

3.4 Tahap Penyelesaian

Setelah selesai melakukan perancangan alat-alat, langkah selanjutnya adalah perakitan. Tahap perakitan dimulai dengan urutan sebagai berikut :

1. Merangkai komponen elektronik

Komponen elektronik, minimum sistem AT89S51, sensor ultrasonik, LCD dan *speaker* dirangkai sesuai dengan perancangan yang telah dibuat. Komponen dipasang pada tempatnya sesuai dengan *layout* PCB.

2. Memasang PCB ke dalam box

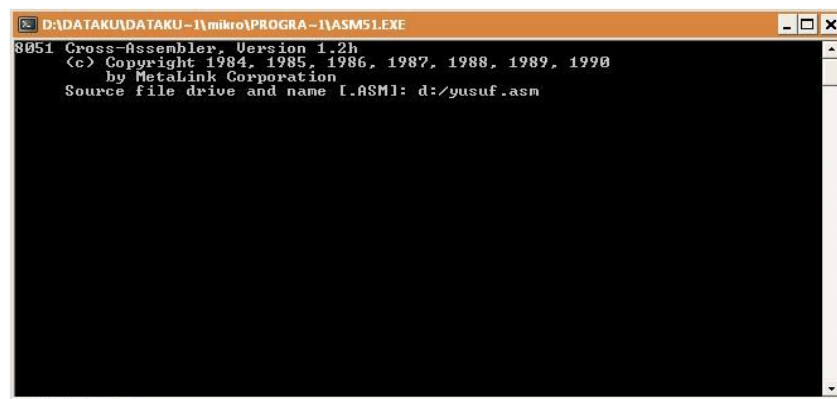
PCB yang sudah dipasangi komponen elektronik dan komponen mikrokontroler dipasang ke dalam box agar lebih rapi dan teratur.

3. Pemrograman mikrokontroler AT89S51

Pemrograman dilakukan setelah semua komponen elektronika dan komponen mikrokontroler terpasang dengan benar. Pemrograman dilakukan dengan menggunakan bahasa *Assembler*. Listing program ditulis dengan menggunakan program Notepad dan *file* disimpan dengan ekstensi “.asm”. Kemudian *file* “.asm” tersebut di- *load* dengan program *compiler* ASM51

untuk dirubah menjadi *file* “.hex”. Setelah *file* dirubah menjadi *file* “.hex” kemudian di-load dengan menggunakan program *compiler* AEC ISP. Tujuannya adalah untuk memasukkan program mikro ke dalam *downloader* mikrokontroler AT89S51. Untuk proses pemrograman ke IC AT89S51 ditunjukkan seperti di bawah ini :

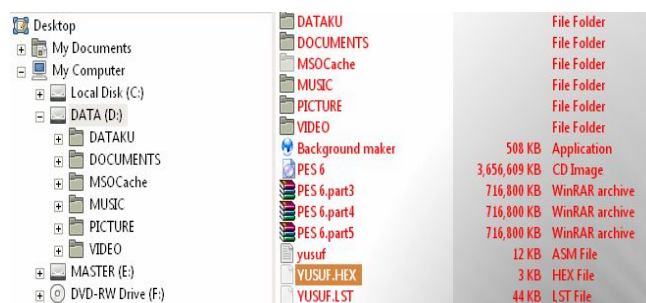
- Program ditulis dengan menggunakan Notepad kemudian file disimpan dengan nama yusuf.asm
- Dibuka program *compiler* ASM51 untuk merubah file yusuf.asm menjadi yusuf.hex.



Gambar 3.9 Load program yusuf.asm

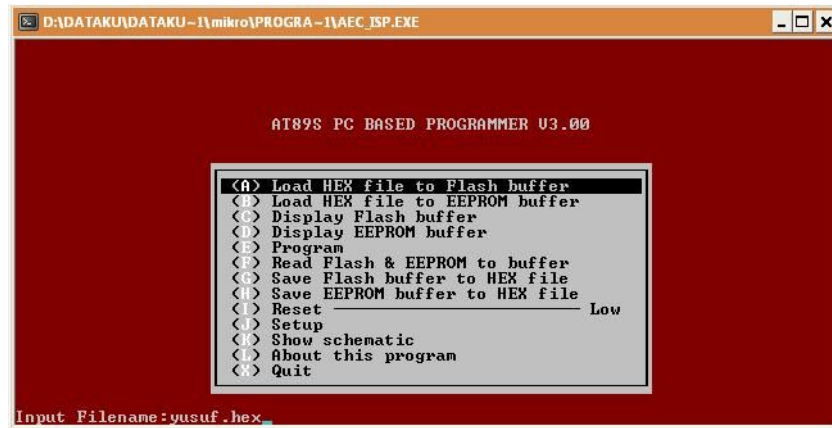


Gambar 3.10 File yusuf.asm



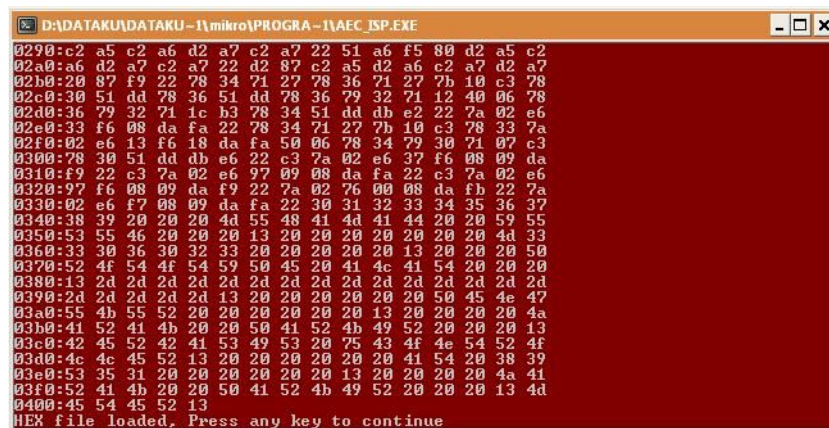
Gambar 3.11 File yusuf.hex

- c. Dibuka program *downloader* AEC ISP seperti terlihat pada gambar dibawah ini :



Gambar 3.12 Proses *load* file hex

Kemudian memilih pada bagian A lalu memasukkan nama program yang akan *download*. Nama filenya adalah yusuf.hex. Proses akan berlanjut dengan inisialisasi memori program seperti terlihat pada gambar di bawah ini :

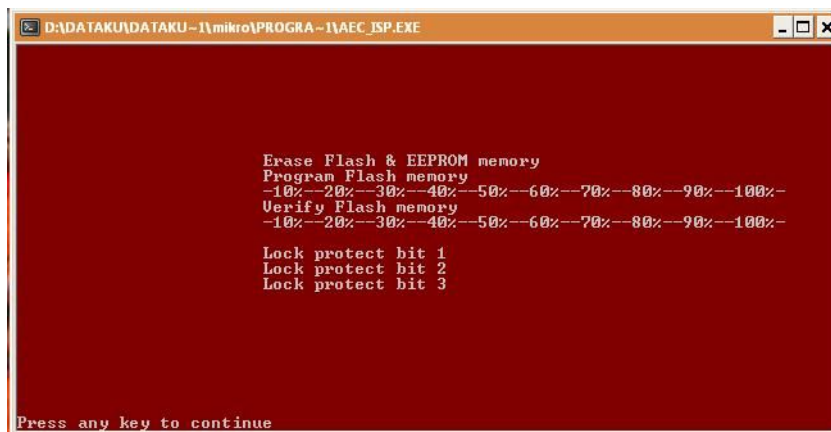


Gambar 3.13 Proses inisialisasi memori program

- d. Proses selanjutnya adalah *download* program ke IC AT89S51 dengan memilih pada bagian E.



Gambar 3.14 *Download* program ke IC AT89S51



Gambar 3.15 Proses *download* program ke IC berhasil

4. *Finishing*

Setelah semuanya terpasang dengan baik, maka tahap selanjutnya adalah tahap *finishing* dengan merapikan kabel-kabel dan merapikan box.

5. Ujicoba

Setelah terpasang menjadi sebuah prototipe sensor parkir mobil dengan baik, maka dilakukan ujicoba. Ujicoba dilakukan dengan melakukan tes untuk mengukur jarak yang telah ditentukan. Jarak yang diukur divariasikan berdasarkan tingkat presisi dan keakuratan alat.

BAB IV

IMPLEMENTASI DAN ANALISA

4.1 Uji Coba Alat

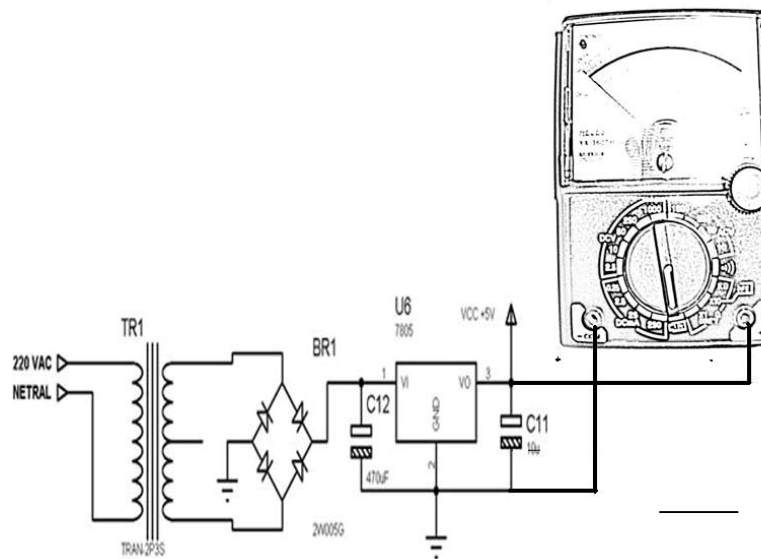
Setelah pembuatan Prototipe Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51 selesai, tahap berikutnya adalah proses pengujian dan pembahasan tentang kinerja dari alat ini. Pengujian ini dilakukan untuk mengetahui cara kerja dan fungsi dari masing-masing komponen utama serta mengetahui cara pengoperasian dari alat ini.

4.1.1 Pengujian Rangkaian Catu Daya

Pengujian rangkaian catu daya dilakukan dengan tahapan sebagai berikut :

1. Panel penunjuk multimeter diarahkan pada Volt DC.
2. Multimeter diatur nilainya sesuai dengan tegangan yang akan diukur.
3. Kabel merah pada multimeter dihubungkan dengan kutub positif trafo dan kabel hitam dihubungkan dengan kutub negatif trafo.
4. Jika jarum pada multimeter menunjukkan nilai yang tepat maka trafo dalam keadaan baik.

Pada rangkaian ini menggunakan trafo *step down*. Tegangan listrik rumah sebesar 220 Volt diubah menjadi 5 Volt. Besaran tegangan yang dihasilkan trafo tersebut masih merupakan tegangan AC dan kemudian diubah menjadi tegangan DC menggunakan rangkaian penyearah. Rangkaian tersebut menggunakan empat buah dioda, digunakan dioda karena kemampuan dioda yang hanya mengalirkan arus searah.



Gambar 4.1 Pengujian rangkaian catu daya

4.1.2 Pengujian Rangkaian Mikrokontroler

Pengujian rangkaian mikrokontroler dilakukan dengan langkah sebagai berikut :

1. Dibuat rangkaian LED dengan anoda ke +5V dan ditambah dengan resistor 220 Ω .
2. IC AT89S51 diberi tegangan +5V
3. LED dihubungkan dengan P1.0
4. Dibuat program dengan bahasa *assembler*. Pada port P1.0 diberikan nilai *low*.

Listing programnya sebagai berikut :

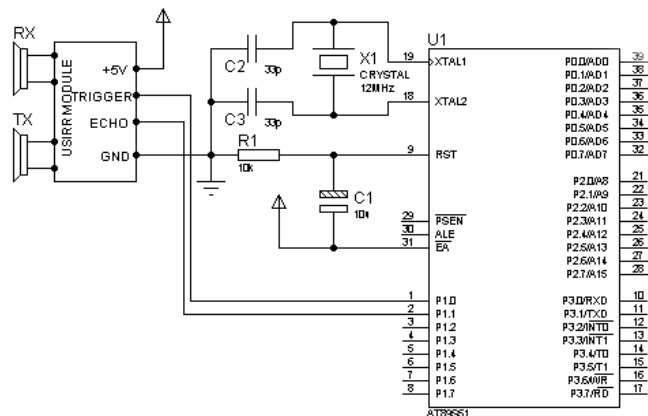
Start : Clr P1.0

 Jmp Start

End

5. Jika LED menyala maka IC AT89S51 dalam keadaan baik.

Berikut gambar pengujian rangkaiannya :



Gambar 4.3 Pengujian rangkaian sensor ultrasonik SRF04

Pengujian rangkaian sensor ultrasonik SRF04 dilakukan dengan menghubungkan antara modul sensor ultrasonik SRF04 dengan rangkaian mikrokontroler AT89S51. Pin – pin pada SRF04 yang dihubungkan antara lain pin sumber tegangan +5v dihubungkan dengan kutub positif trafo regulator +5v, Pin *Trigger* dihubungkan dengan P1.0, Pin *Echo* dihubungkan dengan P1.1 dan pin *Ground* dihubungkan dengan kutub negatif trafo regulator.

Cara kerja dari sensor ultrasonik SRF04 adalah mula – mula SRF04 diaktifkan melalui pin *Trigger* minimal 10 µs dengan mengirimkan pulsa positif dari IC mikrokontroler. Selanjutnya pin TX akan mengirim sinyal pada saat logika 1 atau *high* yang mengenai penghalang dan sinyal pantulan dari penghalang akan diterima oleh RX. Pada saat menerima sinyal pantulan, RX berlogika 0 atau *low*, dimana sinyal dari RX akan dilewatkan melalui pin *Echo*. Lebar sinyal dari *Echo* inilah yang akan digunakan untuk pengukuran jarak. Selanjutnya adalah melakukan ujicoba pengukuran jarak sensor ultrasonik SRF04 dengan cara menempatkan sensor ultrasonik di depan penghalang dan memvariasi jarak pengukuran. Hasil yang didapat dari pengukuran jarak adalah sebagai berikut :

Tabel 4.1 Hasil pengukuran jarak dengan sensor ultrasonik SRF04

Jarak penghalang (cm)	Jarak yang tampil pada LCD (m)
0	0,03 = 3 cm
1	0,03 = 3 cm
2	0,03 = 3 cm

3	0,04 = 4 cm
4	0,05 = 5 cm
5	0,06 = 6 cm
6	0,07 = 7 cm
7	0,08 = 8 cm
8	0,10 = 10 cm
9	0,11 = 11 cm
10	0,12 = 12 cm
50	0,58 = 58 cm
100	1,1 = 110 cm
200	2,31 = 231 cm
300	3,47 = 347 cm
350	2,50 = 250 cm
400	2,91 = 291 cm

Tabel 4.2 Hasil perbandingan pengukuran jarak

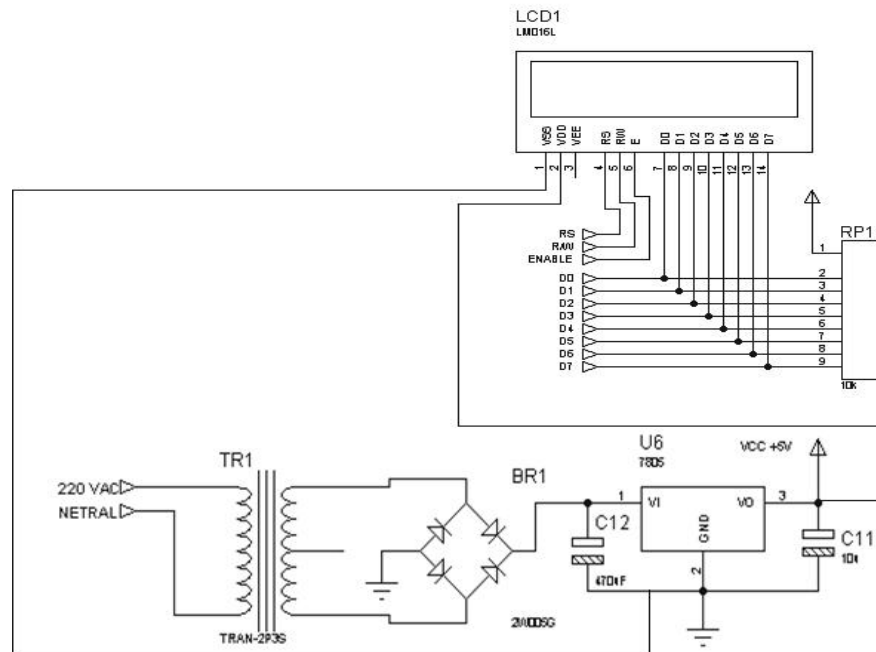
Pengukuran jarak menggunakan meteran (cm)	Pengukuran jarak menggunakan sensor ultrasonik (cm)	Selisih (cm)
0	3	3
10	12	2
20	24	4
30	35	5
40	46	6
50	58	8

60	69	9
70	78	8

Dari pengujian yang dilakukan didapatkan hasil bahwa sensor ultrasonik SRF04 mampu mengukur jarak mulai dari 0 cm sampai dengan 300 cm. Pada pengukuran jarak 350 cm dan 400 cm didapatkan hasil yang tidak sesuai atau menyimpang jauh dengan jarak sebenarnya, hal ini disebabkan karena kemampuan sensor ultrasonik SRF04 hanya mampu mengukur jarak maksimum 300 cm atau 3 meter. Angka yang ditampilkan pada LCD terdapat selisih dengan jarak sebenarnya, hal ini dipengaruhi oleh beberapa faktor yang dapat mengurangi keakuratan pengukuran jarak dengan menggunakan sensor ultrasonik SRF04 diantaranya adalah karakteristik penghalang atau objek yang dapat memantulkan sinyal kembali ke sensor ultrasonik dan adanya sinar matahari yang dapat mengganggu jalannya sinyal dari sensor ultrasonik. Secara teori, sensor ultrasonik SRF04 ini dapat bekerja dengan baik sesuai dengan spesifikasi dari sensor ultrasonik SRF04 yaitu dapat mengukur jarak mulai dari jarak minimum 3cm sampai dengan jarak maksimum 3 m.

4.1.4 Pengujian Rangkaian LCD

Tujuan pengujian rangkaian LCD adalah untuk mengecek apakah LCD bekerja dengan baik. Untuk mengetahui apakah LCD berfungsi dengan baik atau tidak, bisa dilakukan dengan menghubungkannya dengan catu daya yang diberi tegangan 5 Volt. Rangkaian yang digunakan adalah sebagai berikut :

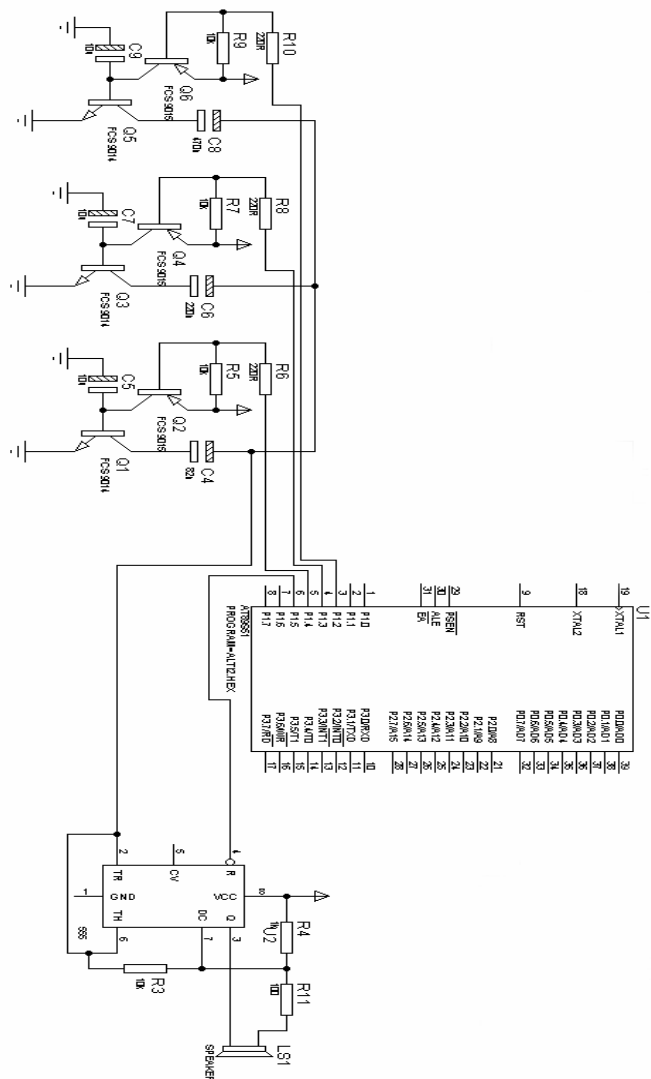


Gambar 4.4 Pengujian rangkaian LCD

Pin – pin pada LCD yang dihubungkan antara lain pin VSS dihubungkan dengan *ground* pada catu daya dan VDD dihubungkan dengan kutub positif +5v pada catu daya. Untuk mengatur tingkat kecerahan atau kontras dari LCD dilakukan dengan cara menghubungkan pin VEE dengan trimpot. Setelah rangkaian tersebut dihubungkan dengan arus listrik, LCD dapat menyala dengan baik dan layak digunakan.

4.1.5 Pengujian Rangkaian *Speaker*

Pengujian rangkaian *speaker* dimaksudkan untuk mengecek apakah *speaker* bekerja dengan baik. Untuk mengetahui apakah *speaker* bisa berfungsi dengan baik, dapat dilakukan dengan rangkaian sebagai berikut :



Gambar 4.5 Pengujian *speaker*

Pengujian *speaker* dapat dilakukan dengan menghubungkan kutub positif dan kutub negatif pada *speaker* dengan baterai 3 volt. Setelah dicek *speaker* masih mengeluarkan bunyi. Sehingga *speaker* dalam keadaan baik.

4.1.6 Pengujian Rangkaian Keseluruhan

Pengujian rangkaian keseluruhan dilakukan setelah pengecekan mulai dari bagian masing-masing rangkaian penyusun dan pengisian program ke dalam IC mikrokontroler AT89S51 selesai. Adapun langkah-langkah pengujiannya adalah sebagai berikut :

1. Menghubungkan kabel dari rangkaian ke colokan listrik.
2. Mengatur jarak penghalang yang digunakan untuk pengukuran.
3. Mencatat hasil pengukuran untuk kemudian dianalisa.



Gambar 4.6 Prototipe sensor parkir mobil

Sensor ultrasonik bekerja berdasarkan prinsip pantulan gelombang suara, dimana sensor menghasilkan gelombang suara yang kemudian menangkapnya kembali dengan perbedaan waktu sebagai dasar pengindraannya. Perbedaan waktu antara gelombang suara yang dipancarkan dan yang diterima kembali adalah berbanding lurus dengan jarak atau tinggi objek yang memantulkannya.

Pengujian alat dilakukan dengan mengatur jarak penghalang yang ada di depan sensor. Hasil yang didapat adalah seperti pada tabel 4.3 di bawah ini :

Tabel 4.3 Hasil pengujian alat

Jarak penghalang (cm)	Jarak yang tampil pada LCD (m)	Bunyi <i>speaker</i>
0	0,03	Tinggi
1	0,03	Tinggi
2	0,03	Tinggi
3	0,04	Tinggi
4	0,05	Tinggi
5	0,06	Tinggi

6	0,07	Tinggi
7	0,08	Tinggi
8	0,10	Tinggi
9	0,11	Tinggi
10	0,12	Tinggi
11	0,13	Tinggi
12	0,14	Tinggi
13	0,16	Tinggi
14	0,17	Tinggi
15	0,19	Tinggi
16	0,20	Tinggi
17	0,20	Tinggi
18	0,22	Tinggi
19	0,23	Tinggi
Jarak penghalang (cm)	Jarak yang tampil pada LCD (m)	Bunyi <i>speaker</i>
20	0,24	Tinggi
21	0,25	Sedang
22	0,27	Sedang
23	0,28	Sedang
24	0,29	Sedang
25	0,30	Sedang
26	0,31	Sedang

27	0,32	Sedang
28	0,34	Sedang
29	0,35	Sedang
30	0,35	Sedang
31	0,36	Sedang
32	0,37	Sedang
33	0,39	Sedang
34	0,40	Sedang
35	0,41	Sedang
36	0,42	Sedang
37	0,43	Sedang
38	0,44	Sedang
39	0,45	Sedang
40	0,46	Sedang
Jarak penghalang (cm)	Jarak yang tampil pada LCD (m)	Bunyi <i>speaker</i>
41	0,48	Sedang
42	0,49	Sedang
43	0,50	Rendah
44	0,51	Rendah
45	0,52	Rendah
46	0,53	Rendah
47	0,54	Rendah

48	0,55	Rendah
49	0,57	Rendah
50	0,58	Rendah
51	0,59	Rendah
52	0,60	Rendah
53	0,61	Rendah
54	0,63	Rendah
55	0,64	Rendah
56	0,65	Rendah
57	0,66	Rendah
58	0,67	Rendah
59	0,68	Rendah
60	0,69	Rendah
61	0,70	Rendah
Jarak penghalang (cm)	Jarak yang tampil pada LCD (m)	Bunyi <i>speaker</i>
62	0,71	Rendah
63	0,72	Rendah
64	0,73	Rendah
65	0,75	Rendah
66	0,76	Rendah
67	0,77	Rendah
68	0,78	Rendah

69	0,79	Rendah
70	0,79	Rendah

Dari pengujian alat yang dilakukan didapatkan hasil pengukuran yang mempunyai selisih dengan jarak sebenarnya. Karakteristik objek penghalang berpengaruh pada hasil pengukuran jarak dengan menggunakan sensor ultrasonik SRF04. Range jarak yang digunakan untuk membedakan karakter bunyi dari *speaker* yaitu 0 cm – 25 cm (tinggi), 25 cm – 50 cm (sedang) dan 50 cm – 70 cm (rendah). Bunyi *speaker* yang berbeda – beda setiap rangenya dipengaruhi oleh resistor digunakan. Jika resistor yang digunakan mempunyai nilai hambatan yang besar, maka bunyi *speaker* rendah. Tetapi jika resistor yang digunakan mempunyai nilai hambatan yang kecil, maka bunyi *speaker* semakin tinggi.

Hasil pengujian dapat membuktikan bahwa sensor ultrasonik bekerja berdasarkan kemampuan penghalang memantulkan kembali gelombang ultrasonik yang dikirim oleh sensor ultrasonik, gangguan pada pendeteksiaan sensor ultrasonik dapat diakibatkan oleh karakteristik penghalang yang kurang mampu untuk memantulkan gelombang bunyi dengan baik dan adanya interferensi gelombang dengan frekuensi yang sama.

BAB V

PENUTUP

5.1 Kesimpulan

Dari keseluruhan proses pembuatan tugas akhir ini, dapat disimpulkan sebagai berikut :

1. Mikrokontroler AT89S51 berfungsi sebagai pengendali utama pada pemrosesan data jarak parkir yang dihasilkan dari sensor ultrasonik.
2. Penggunaan modul sensor ultrasonik SRF04 dapat menghasilkan data yang lebih akurat karena rangkaian TX dan RX sudah menjadi satu pada modul. Sedangkan penggunaan sensor ultrasonik yang biasa, data yang diperoleh kurang akurat karena harus membuat rangkaian TX dan RX sendiri. Jika rangkaianannya salah sedikit maka data yang didapat kurang akurat.
3. Jarak parkir ditampilkan melalui LCD dengan satuan ukur meter.
4. Speaker dapat digunakan sebagai indikator suara pada sensor parkir mobil.

5.2 Saran

Dari hasil penelitian dalam pembuatan Prototipe Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51, maka penulis memberikan saran kepada pembaca dalam rangka kemajuan alat ini ke depan, diantaranya :

1. Pengembangan model Prototipe Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51. Dimana alat yang dibuat tidak hanya dalam bentuk prototipe, tetapi langsung diterapkan pada mobil.
2. Penggunaan saklar untuk mempermudah saat menghidupkan dan mematikan alat.

DAFTAR PUSTAKA

- Putra, Agfianto Eko, 2005. *Belajar Mikrokontroler AT89C51/52/53 (Teori dan Aplikasi)*, Yogyakarta : Gava Media.
- Prihono, 2009. *Jago Elektronika Secara Otodidak*, Jakarta : Kawan Pustaka.
- Txman, 2007. [Http://elektronika21.blogspot.com/2007/11/mikrokontroler-at89s51.html](http://elektronika21.blogspot.com/2007/11/mikrokontroler-at89s51.html) diakses tanggal 27 Desember 2008
- Triwiyanto, 2006. [Http://mytutorialcafe.com/mikrokontroller/mikrokontroller-dasar.html](http://mytutorialcafe.com/mikrokontroller/mikrokontroller-dasar.html) diakses tanggal 11 Mei 2009
- Soebhakti, Hendawan, 2007. [Http://www.polibatam.ac.id](http://www.polibatam.ac.id) diakses tanggal 27 Desember 2008
- Anonim, 2007. [Http://www.toko-elektronika.com/tutorial/uc2.html](http://www.toko-elektronika.com/tutorial/uc2.html) diakses tanggal 27 Desember 2008
- Anonim, 2009. [Http://indomicron.co.cc/elektronika/analog/pengertian-sensor/](http://indomicron.co.cc/elektronika/analog/pengertian-sensor/) diakses tanggal 25 Mei 2009
- Anonim, 2009. [Http://digilib.petra.ac.id/prototype-chapter3.pdf](http://digilib.petra.ac.id/prototype-chapter3.pdf) diakses tanggal 21 April 2009
- Anonim, 2009. [Http://roboticunpad.blogspot.com/2009/04/sensor.html](http://roboticunpad.blogspot.com/2009/04/sensor.html) diakses tanggal 7 Juli 2009

LAMP IRAN

Dokumentasi Alat



Listing Program Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51

```
$mod51
;=====
;      Prototype Alat Pengukur Jarak Parkir Berbasis Mikrokontroler AT89S51
;      Nama       : Muhamad Yusuf
;      NIM        : M3306023
;=====
LCD_DT EQU P0           ;JALUR DATA BUS LCD
RS      BIT P2.5         ;PIN REGISTER SELECT(COMMAND/DATA)
RW      BIT P2.6         ;PIN READ/WRITE
EN      BIT P2.7         ;PIN ENABLE COMMAND/DATA

SIGNAL  BIT P1.0         ;PIN I/O PULSA START MODUL USIRR DAN REVERSE ECHO
                ;PULSE
ECHO    BIT P1.1         ;PIN REVERSE ECHO PULSE (UTK MODUL SRF-04 P1.0
                ;DIGANTI P1.4)

m2      EQU 0CEH-4       ;ALAMAT KARAKTER 'm',meter
m1      EQU 0C9H-4       ;ALAMAT KARAKTER ',','koma
SAT      EQU 0CCH-4      ;ALAMAT DATA TAMPILAN SATUAN KE LCD
PUL      EQU 0CBH-4      ;ALAMAT DATA TAMPILAN PULUHAN KE LCD
RAT      EQU 0CAH-4      ;ALAMAT DATA TAMPILAN RATUSAN KE LCD
RIB      EQU 0C8H-4      ;ALAMAT DATA TAMPILAN RIBUAN KE LCD
PULRIB   EQU 0C7H-4      ;ALAMAT DATA TAMPILAN PULUHAN RIBU KE LCD

        DSEG
        org      30h
SizeX    equ 2      ;16 bit!

;***** Byte Ordering : LS Byte .... MS Byte
Operand:  DS      SizeX
Pembagi:   DS      SizeX
HasilBagi: DS      SizeX
SisaBagi:  DS      SizeX
Pengali    EQU Pembagi
HasilKali  EQU      HasilBagi

FACMUL:    DS 1
FACDIV:    DS 1

BUFCMPLO:  DS 1
BUFCMPHI:  DS 1

        CSEG
        ORG 00H
        AJMP START

        ;ORG 30H
START:
DG5      EQU 43H ;ALAMAT VEKTOR DIGIT KE 5
DG4      EQU 44H ;ALAMAT VEKTOR DIGIT KE 4
DG3      EQU 45H ;ALAMAT VEKTOR DIGIT KE 3
DG2      EQU 46H ;ALAMAT VEKTOR DIGIT KE 2
DG1      EQU 47H ;ALAMAT VEKTOR DIGIT KE 1

        MOV      TMOD,#11H      ;TIMER0/1 16BIT
        MOV      TH0,#00H
        MOV      TL0,#00H

        CLR      TF0            ;NOL-KAN FLAG TIMER 0
        SETB     EA            ;AKTIFKAN SEMUA INTERUPSI

        CALL     INITLCD ;INISIALISASI LCD
;-----
ACALL    LINE1      ;MULAI TULIS DARI AWAL BARIS ATAS LCD
MOV      DPTR,#LABEL11
CALL     DISP_STRING ;TULIS KARAKTER KE LCD
```

```

        ACALL    LINE2            ;MULAI TULIS DARI AWAL BARIS BAWAH LCD
        MOV      DPTR,#LABEL12
        CALL     DISP_STRING
        CALL     DELAY_1S         ;TULIS KARAKTER KE LCD
        ;-----
        ACALL    LINE1            ;MULAI TULIS DARI AWAL BARIS ATAS LCD
        MOV      DPTR,#LABEL21
        CALL     DISP_STRING      ;TULIS KARAKTER KE LCD

        ACALL    LINE2            ;MULAI TULIS DARI AWAL BARIS BAWAH LCD
        MOV      DPTR,#LABEL22
        CALL     DISP_STRING      ;TULIS KARAKTER KE LCD
        CALL     DELAY_1S
        ;-----
        ACALL    LINE1            ;MULAI TULIS DARI AWAL BARIS ATAS LCD
        MOV      DPTR,#LABEL31
        CALL     DISP_STRING      ;TULIS KARAKTER KE LCD

        ACALL    LINE2            ;MULAI TULIS DARI AWAL BARIS BAWAH LCD
        MOV      DPTR,#LABEL32
        CALL     DISP_STRING      ;TULIS KARAKTER KE LCD
        CALL     DELAY_1S
        ;-----
        ACALL    LINE1            ;MULAI TULIS DARI AWAL BARIS ATAS LCD
        MOV      DPTR,#LABEL41
        CALL     DISP_STRING      ;TULIS KARAKTER KE LCD

        ACALL    LINE2            ;MULAI TULIS DARI AWAL BARIS BAWAH LCD
        MOV      DPTR,#LABEL42
        CALL     DISP_STRING      ;TULIS KARAKTER KE LCD
        CALL     DELAY_1S
        ;-----

        CALL     CLR_LCD

        CALL     CLEAR_ALARM

ULANG:
        MOV      P1,#0FFH
        CALL     SIG_START

HOLD1:  JNB      ECHO,HOLD1

        MOV      TL0,#0
        MOV      TH0,#0
        CLR      TF0
        SETB     TR0

HOLD2:  JB       ECHO,HOLD2

        CLR      TR0

        JBC      TF0,OVER2
        JMP      HASIL2
OVER2:  MOV      TL0,#0FEH
        MOV      TH0,#0FFH
HASIL2:
        MOV      FACMUL,#3
        MOV      FACDIV,#15
        CALL     KALI_BAGI
        ACALL    DELAY_1S

        MOV      P1,#0FFH

        ACALL    LINE1
        MOV      DPTR,#JARAK
        CALL     DISP_STRING

```

```

MOV A,#m2
ACALL COMMAND
MOV DPTR,#METER
;MOV A,'#M' ;DISPLAY KARAKTER METER (m)
ACALL DISP_STRING

MOV A,#m1
ACALL COMMAND
MOV A,'#',' ' ;DISPLAY KARAKTER , (koma)
ACALL DDISP

MOV BUFCMPLO,TL0
MOV BUFCMPHI,TH0
CALL CEK_ALARM
ACALL AMBIL_DATA
CALL DELAY_1S
JMP ULANG

CLEAR_ALARM:
CLR P2.0
CLR P2.1
CLR P2.2
CLR P2.3
RET
CEK_ALARM:
PUSH ACC
;PUSH B
;PUSH 0
;PUSH 1
;clr cY

mov operand,BUFCMPLO
mov operand+1,BUFCMPHI
mov pembagi,#1
mov pembagi+1,#0
call PERKALIAN

mov operand,HASILBAGI ;t10
mov operand+1,HASILBAGI+1 ;th0
mov pembagi,#LOW 250
mov pembagi+1,#HIGH 250
call PEMBAGIAN
mov A,HASILBAGI

CJNE A,#0,L0
CALL LED1ON ;ALARM MODE1 JARAK SANGAT DEKAT
JMP OUT
L0: CJNE A,#1,L1
CALL LED2ON ;ALARM MODE2 JARAK AGAK DEKAT
JMP OUT
L1: CJNE A,#2,L2
CALL LED3ON ;ALARM MODE3 JARAK AGAK JAUH
JMP OUT
L2: CJNE A,#3,L3
CALL LED4ON ;ALARM MODE4 JARAK SANGAT JAUH
L3: CALL LED4ON

;POP 1
;POP 0
;POP B
OUT: POP ACC
RET

;0000H-012CH ---->jarak dekat 0000 0000 0000 0000 - 0000 0001 0010 1100
;012DH-0258H ---->jarak menengah 0000 0010 0101 1000 - 0000 0001 0010 1101
;0259H-(~) ---->jarak jauh 0000 0010 0101 1001 - ~

LED1ON:SETB P2.0
CLR P2.1

```

```

        CLR    P2.2
        CLR    P2.3
        RET

LED2ON: CLR    P2.0
        SETB   P2.1
        CLR    P2.2
        CLR    P2.3
        RET

LED3ON: CLR    P2.0
        CLR    P2.1
        SETB   P2.2
        CLR    P2.3
        RET

LED4ON: CLR    P2.0
        CLR    P2.1
        CLR    P2.2
        SETB   P2.3
        RET

CLRLCD: MOV    A, #1
        CALL   COMMAND
        RET

LINE1:  MOV    A, #80H
        CALL   COMMAND
        RET

LINE2:  MOV    A, #0C0H
        CALL   COMMAND
        RET

SIG_START:
        CLR    SIGNAL
        CALL   DL_10uS
        SETB   SIGNAL
TILL_STRONG:
        NOP
        NOP
        JNB    SIGNAL, TILL_STRONG ;TUNGGU SAMPAI LOGIKA "1" TERCAPAI PENUH
        RET

;DELAY 20 uS
DL_10uS:
        NOP ;NO OPERATION 1uS
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        RET

DELAY_1S:
        PUSH   7
        MOV    R7, #10
NX_1S:  ACALL  DLY1
        DJNZ   R7, NX_1S
        POP    7
        RET

DLY1:   MOV    R4, #200
DL1:    PUSH   4
        MOV    R4, #250

```

```

        DJNZ  R4,$
        POP   4
        DJNZ  R4,DL1
        ret

KALI_BAGI:
        mov  operand,tl0      ;low
        mov  operand+1,th0    ;high
        mov  pembagi,FACMUL ;3   ;low
        mov  pembagi+1,#0     ;high
        call perkalian
        mov  operand,hasilbagi ;low
        mov  operand+1,hasilbagi+1;high
        mov  pembagi,FACDIV   ;15 ;low
        mov  pembagi+1,#0     ;high
        call pembagian

        mov  tl0,hasilbagi
        mov  th0,hasilbagi+1
        RET

;=====
;          KONVERSI HEXA KE DECIMAL (PENGAMBILAN DATA)
;=====

AMBIL_DATA:
        ;MOV
        DPL,TL0
        ;MOV
        DPH,TH0

DIGIT5:
        MOV
        DG5,#00H
        CLR  C

LOOP4:
        MOV  A,TL0 ;DPL
        SUBB A,#10H;0001 0000
        MOV  20H,A
        ;-----
        MOV  A,TH0 ;DPH
        SUBB A,#27H;0010 0111
        MOV  21H,A
        ;-----
        JBC  CY,DIGIT4
        INC  DG5
        MOV  TL0,20H ;DPL,20H
        MOV  TH0,21H ;DPH,21H
        SJMP LOOP4

DIGIT4:
        MOV  DG4,#00H

LOOP5:
        MOV  A,TL0 ;DPL
        SUBB A,#0E8H;1110 1000
        MOV  20H,A
        ;-----
        MOV  A,TH0 ;DPH
        SUBB A,#03H;0000 0011
        MOV  21H,A
        ;-----
        JBC  CY,DIGIT3
        INC  DG4
        MOV  TL0,20H ;DPL,20H
        MOV  TH0,21H ;DPH,21H
        SJMP LOOP5

DIGIT3:
        MOV  DG3,#00H

LOOP6:
        MOV  A,TL0 ;DPL
        SUBB A,#64H;0110 0100
        MOV  20H,A

```

```

;-----
MOV A,TH0 ;DPH
SUBB A,#00H;0000 0000
MOV 21H,A
;-----
JBC CY,DIGIT2
INC DG3
MOV TL0,20H ;DPL,20H
MOV TH0,21H ;DPH,21H
SJMP LOOP6
DIGIT2:
MOV A,TL0 ;DPL
MOV B,#0AH
DIV AB
MOV DG2,A
DIGIT1:
MOV DG1,B
;MOV DPL,DG2
;MOV DPH,DG3
;MOV 40H,DG4

;CALL DISPLAY

;RETI

;FFFFH =65535
;2710 =10000
;03E8 =1000
;0064 =100
;000A =10
;=====
;
; DISPLAY
;=====
DISPLAY:PUSH ACC

MOV A,DG5
CJNE A,#0,HABIS
SETB 10H
MOV A,DG4
CJNE A,#0,HABIS
SETB 11H
MOV A,DG3
CJNE A,#0,HABIS
SETB 12H
MOV A,DG2
CJNE A,#0,HABIS
SETB 13H

HABIS: MOV DPTR,#ASCII

;MOV A,#SAT ;DISPLAY DIGIT SATUAN
;ACALL COMMAND
;MOV A,DG1
;MOVC A,@A+DPTR
;ACALL DDISP

JB 13H,BLANK20
CLR 13H
BLANK2:MOV A,#PUL ;DISPLAY DIGIT PULUHAN
ACALL COMMAND
MOV A,DG2
MOVC A,@A+DPTR
ACALL DDISP
JB 12H,BLANK30
CLR 12H
BLANK3:MOV A,#RAT ;DISPLAY DIGIT RATUSAN
ACALL COMMAND
MOV A,DG3
MOVC A,@A+DPTR

```

```

        ACALL  DDISP

        JB  11H,BLANK40
        CLR  11H
BLANK4: MOV    A,#RIB  ;DISPLAY DIGIT RIBUAN
        ACALL  COMMAND
        MOV    A,DG4
        MOVC   A,@A+DPTR
        ACALL  DDISP

        JB  10H,BLANK50
        CLR  10H
BLANK5: MOV    A,#PULRIB ;DISPLAY DIGIT PULUHAN RIBU
        ACALL  COMMAND
        MOV    A,DG5
        MOVC   A,@A+DPTR
        ACALL  DDISP

        MOV    22H,#0
        POP    ACC
        RET

BLANK20:
        MOV    DG2,#0
        JMP    BLANK2
BLANK30:
        MOV    DG3,#0
        JMP    BLANK3
BLANK40:
        MOV    DG4,#0
        JMP    BLANK4
BLANK50:
        MOV    DG5,#' '
        JMP    BLANK5

INITLCD:
        MOV    A,#38H    ;2 line 5x7
        ACALL  COMMAND
        MOV    A,#0CH    ;lcd on cursor on
        ACALL  COMMAND
        MOV    A,#01H    ;clr lcd
        ACALL  COMMAND
        MOV    A,#06H    ;shift cursor right
        ACALL  COMMAND
        RET

DISP_STRING:
        MOV    A,#0
        MOVC   A,@A+DPTR
        CJNE   A,#13H,LJT_STR
        RET
LJT_STR:
        ACALL  DDISP
        INC    DPTR
        SJMP   DISP_STRING

COMMAND:
        ACALL  BUSY    ;chk lcd is ready to accept command
        MOV    LCD_DT,A
        CLR    RS    ;rs=0 for command
        CLR    RW ;wr=0 for write
        SETB   EN
        CLR    EN
        RET

DDISP: ACALL  BUSY
        MOV    LCD_DT,A
        SETB   RS    ;rs=1 for data
        CLR    RW ;wr=0 for write
        SETB   EN

```



```

        CLR    EN
        RET

BUSY:   SETB   LCD_DT.7
        CLR    RS    ;rs=0 command
        SETB   RW    ;wr=1 read
BACK:   CLR    EN    ;enable
        SETB   EN
        JB     LCD_DT.7,BACK
        RET

;
; ***** Multi Byte divider
;
Pembagian:
        MOV    R0,#HasilBagi
        ACALL  HapusNilai
        MOV    R0,#SisaBagi
        ACALL  HapusNilai
;
        MOV    R3,#SizeX*8
LoopPembagian:
        CLR    C
        MOV    R0,#Operand
        ACALL  GeserKirilX
        MOV    R0,#SisaBagi
        ACALL  GeserKirilX
;
        MOV    R0,#SisaBagi
        MOV    R1,#Pembagi
        ACALL  Perbandingan    ;SisaBagi-Pembagi?
        JC     JanganDikurangi ;SisaBagi<Pembagi, skip!
;
        MOV    R0,#SisaBagi
        MOV    R1,#Pembagi
        ACALL  Pengurangan     ;SisaBagi:=SisaBagi-Pembagi
JanganDikurangi:
        CPL    C
        MOV    R0,#HasilBagi    ;Simpan hasil
        ACALL  GeserKirilX
        DJNZ   R3,LoopPembagian
        RET

GeserKirilX:
        MOV    R2,#SizeX
LeftShift:
        MOV    A,@R0
        RLC    A
        MOV    @R0,A
        INC    R0
        DJNZ   R2,LeftShift
        RET

;
; ***** Multi Byte Multiplier
;
Perkalian:
        MOV    R0,#HasilKali
        ACALL  HapusNilai
;
        MOV    R3,#SizeX*8
LoopPerkalian:
        CLR    C
        MOV    R0,#Pengali+SizeX-1
        MOV    R2,#SizeX
GeserKanan:
        MOV    A,@R0
        RRC    A
        MOV    @R0,A
        DEC    R0

```

```

        DJNZ    R2,GeserKanan

        JNC     JanganDitambah

        MOV     R0,#HasilKali
        MOV     R1,#Operand
        ACALL   Penambahan
JanganDitambah:
        CLR     C
        MOV     R0,#Operand
        ACALL   GeserKirilX
        DJNZ    R3,LoopPerkalian
        RET

;
; ***** Multi Byte Adder
;
Penambahan:
        CLR     C
        MOV     R2,#SizeX
LoopPenambahan:
        MOV     A,@R0
        ADDC    A,@R1
        MOV     @R0,A
        INC     R0
        INC     R1
        DJNZ    R2,LoopPenambahan
        RET

;
; ***** Multi Byte Comparator
;
Perbandingan:
        CLR     C
        MOV     R2,#SizeX
LoopPerbandingan:
        MOV     A,@R0
        SUBB    A,@R1
        INC     R1
        INC     R0
        DJNZ    R2,LoopPerbandingan
        RET

;
; ***** Multi Byte Subtractor
;
Pengurangan:
        CLR     C
        MOV     R2,#SizeX
LoopPengurangan:
        MOV     A,@R0
        SUBB    A,@R1
        MOV     @R0,A
        INC     R0
        INC     R1
        DJNZ    R2,LoopPengurangan
        RET

;
; ***** Multi Byte Eraser
;
HapusNilai:
        MOV     R2,#SizeX
LoopHapus:
        MOV     @R0,#0
        INC     R0
        DJNZ    R2,LoopHapus
        RET

;
; ***** Multi Byte Copier
;
Copy:
        MOV     R2,#SizeX
LoopCopy:

```

```

MOV    A,@R0
MOV    @R1,A
INC     R0
INC     R1
DJNZ    R2,LoopCopy
RET

```

```

;=====

```

```

ASCII:  DB  30H,31H,32H,33H,34H,35H,36H,37H,38H,39H

```

```

LABEL11: DB '    MUHAMAD  YUSUF    ',13H
LABEL12: DB '          M3306023    ',13H

```

```

LABEL21: DB '    PROTOTYPE ALAT    ',13H
LABEL22: DB '-----',13H

```

```

LABEL31: DB '          PENGUKUR      ',13H
LABEL32: DB '          JARAK  PARKIR    ',13H

```

```

LABEL41: DB 'BERBASIS uCONTROLLER',13H
LABEL42: DB '          AT 89S51        ',13H

```

```

JARAK: DB '          JARAK  PARKIR    ',13H
METER: DB 'METER',13H

```

```

END

```

DATASHEET LCD 2X16 SLCDV 1.2

DATASHEET DEVANTECH SRF04